

VisionLabs LUNA Access

User manual

2.19.0

Contents

1	Glossary	9
2	Introduction	11
3	System requirements	12
4	Supported components and versions	13
4.1	Supported components	13
4.2	Services version matching	14
4.3	Device version matching	15
4.4	Supported versions of controllers and converters	16
5	Working with Access	18
5.1	Access Localization	18
5.2	User roles and permissions	18
5.3	Adding an account	19
5.4	Authorization in Access	19
5.5	Signing out of the Access account	21
5.6	Access sections	22
5.6.1	Creating a component	23
5.6.2	General information about the component	29
5.6.3	Component grouping	30
5.6.4	Editing component	32
5.6.5	Restarting a component	33
5.6.6	Component removal	34
6	Logs	37
6.1	Logs filtering	39
7	Access function	43
7.1	Import settings function	43
7.2	Export settings function	46
7.3	Reset Settings function	47
7.4	Full name variables	47
7.5	Other functions	48
7.5.1	Documentation	48
8	Services	49
8.1	Apacs	49
8.1.1	Apacs functionality	49

8.1.2	Configuring parameters for connecting to the APACS ACS	49
8.2	Bastion	51
8.2.1	Bastion functionality	51
8.2.2	Bastion settings	51
8.3	Bolid	53
8.3.1	Bolid settings	53
8.4	CbsAkbars	54
8.4.1	Configuring parameters for connecting to CbsAkbars	54
8.5	CbsAlpha	56
8.5.1	Setting up parameters for connecting to CbsAlpha	56
8.6	CbsAlphaListSynchronisation	58
8.6.1	Configuring CbsAlphaListSynchronisation Settings	58
8.7	CbsMts	58
8.7.1	Configuring CbsMts settings	58
8.8	CbsVtb	59
8.8.1	Configuring parameters for connecting to CbsVtb	60
8.9	CryptoPro	61
8.9.1	Configuring parameters for connecting to CryptoPro	62
8.10	EyelsProxy	62
8.10.1	Configuring parameters for connecting to EyelsProxy	62
8.11	Gate	63
8.11.1	Configuring Gate settings	63
8.12	Luna	63
8.12.1	Luna settings	64
8.13	LunaAceConverter	65
8.13.1	LuaAceConverter settings	66
8.13.2	Setting up LUNA ACE	66
8.14	LunaCars	66
8.14.1	LunaCars settings	67
8.15	LunaStreams	68
8.15.1	Configuring LunaStreams settings	69
8.16	Parsec	70
8.16.1	Parsec functionality	70
8.16.2	Parsec settings	70
8.17	PercoWeb	71
8.17.1	PercoWeb functionality	71
8.17.2	PercoWeb settings	71
8.18	PersonStorageActualization	73
8.18.1	PersonStorageActualization Settings	73

8.19	Rusguard	74
8.19.1	Rusguard functionality	74
8.19.2	Configuring settings for connecting to Rusguard	74
8.20	Salto	75
8.20.1	Configuring settings for connecting to Salto	75
8.21	Sigur	76
8.21.1	Sigur functions	76
8.21.2	Setting parameters for connecting to Sigur ACS	77
8.22	SigurThroughDatabase	78
8.22.1	LP5 Integration Options	79
8.22.2	Configure settings for connecting to Sigurthdatabase	79
8.23	Strazh	80
8.23.1	Strazh settings	80
8.24	Ubs	82
8.24.1	Ubs settings	82
9	APACS ACS	83
9.1	Supported integration options for APACS ACS	83
9.2	Standard integration using Apacs	84
9.2.1	Guest pass with two-factor authentication	87
9.2.2	Creating a user in RabbitMQ	87
9.3	Methods of interaction with Apacs	88
9.4	Apacs interaction process diagrams	89
9.4.1	Connecting the Apacs service	89
9.4.2	Processing Apacs events with 1 factor	90
9.4.3	Processing Apacs events with 2 factors	91
9.5	Apacs FAQ	92
10	Bastion ACS	94
10.1	Supported integration options for Bastion ACS	94
10.2	Standard integration using Bastion	96
10.3	Setting up the Bastion 3 ACS software	97
10.3.1	Setting up a two-factor Bastion access point	102
10.4	Methods of interaction with Bastion	105
10.5	Bastion Interaction Process Diagrams	106
10.5.1	Connecting the Bastion service and replicating employees	106
10.5.2	Processing Bastion events with 2 factors	108
11	Bolid ACS	110
11.1	Supported integration options for Bolid ACS	110

11.2	Standard integration using Bolid	112
11.3	Setting Bolid ACS	113
11.3.1	Preparatory actions with Orion Pro software	113
11.3.2	Adding an employee in Orion Pro	115
11.3.3	Adding devices to Orion Pro	116
11.3.4	Configuring the “ORION PRO INTEGRATION MODULE” application	119
11.4	Methods of interaction with Bolid	120
11.5	Bolid Interaction Process Diagrams	121
11.5.1	Bolid Service Connection	121
11.5.2	Event processing with 1 factor	123
11.5.3	Event processing with 2 factors	124
11.6	Bolid FAQ	125
12	Gate ACS	126
12.1	Supported integration options for Gate ACS	126
13	Parsec ACS	128
13.1	Supported integration options for Parsec ACS	128
13.2	Standard integration using Parsec	129
13.3	Configuring Access and Parsec ACS integration	131
13.4	Configuring access groups in Parsec ACS	135
13.5	Adding staff to Parsec ACS	136
13.6	Methods of interaction with Parsec	138
13.7	Parsec interaction process diagrams	138
13.7.1	Connecting the Parsec service	138
13.7.2	Parsec event processing with 2 factors	140
14	PERCo-Web ACS	142
14.1	Supported integration options for PERCo-Web ACS	142
14.2	Standard integration using PERCo-Web	143
14.3	Methods of interaction with PERCo-Web	144
14.4	Diagrams of interaction processes with PERCo-Web	145
14.4.1	Connecting the PERCo-Web service	145
14.4.2	PERCo-Web event processing with 1 factor	147
15	Rusguard ACS	149
15.1	Supported integration options for Rusguard ACS	149
15.2	Standard integration using Rusguard	151
15.3	Methods of interaction with Rusguard	152
15.4	Diagrams of interaction processes with RusGuard	153
15.4.1	Diagram of interaction between RusGuard ACS and Access	153

15.4.2	Diagram of interaction between Access and the biometric system	155
16	SALTO ACS	157
16.1	Supported integration options for SALTO ACS	157
16.2	Standard integration using Salto	157
16.3	SALTO ACS Access Levels	159
16.4	Methods for interacting with Salto	159
16.5	SALTO interaction process diagram	160
17	Sigur ACS	163
17.1	Supported integration options for Sigur ACS	163
17.1.1	LP5 Integration Options	163
17.1.2	Integration options with KBS	164
17.1.3	Integration options with LUNA CARS	165
17.2	Standard integrations using Sigur	165
17.3	Setting up Sigur ACS software	168
17.3.1	Setting up access points in Sigur	171
17.3.2	Setting up access modes in the Sigur ACS software	172
17.4	Methods of interaction with Sigur	175
17.5	Diagram of interaction between Sigur ACS and Access	175
17.6	Sigur FAQ	178
18	STRAZH ACS	180
18.1	Supported integration options for STRAZH ACS	180
18.2	Standard integration using STRAZH ACS	181
18.3	Setting up STRAZH ACS software for two-factor authentication	184
18.4	Methods of interaction with STRAZH	185
18.5	STRAZH interaction process diagrams	186
18.5.1	Strazh service connection	186
18.5.2	Modifying employees in STRAZH ACS	188
18.5.3	Processing STRAZH events with 1 factor	189
18.5.4	Processing STRAZH events with 2 factors	190
19	Integrations without ACS	192
20	Controllers	193
20.1	ApacsController	193
20.1.1	Setting up parameters for connecting to the Apacs controller	193
20.2	Gate controller	194
20.2.1	Gate Ethernet-Wiegand converter	194
20.2.2	Gate controller settings	194

20.3	LaurentController	195
20.3.1	Laurent controller settings	195
20.4	PercoController	196
20.4.1	PercooController settings	196
20.5	PusrController	198
20.5.1	Pusr controller settings	198
20.6	Salto controller	199
20.6.1	Salto controller settings	199
20.7	Strazh controller	200
20.7.1	Strazh controller settings	200
21	Devices	202
21.1	Beward	202
21.1.1	Beward settings	202
21.2	BioSmart Quasar	204
21.2.1	BioSmart Quasar settings	204
21.3	Dahua	206
21.3.1	Dahua settings	206
21.4	DahuaThermo	207
21.4.1	DahuaThermo settings	207
21.5	Fortuna315	208
21.5.1	Fortuna315 settings	209
21.6	GrgFaster	210
21.6.1	Configuring settings for connecting to GrgFaster	210
21.7	HikvisionCamera	211
21.7.1	HikvisionCamera settings	211
21.8	HikvisionCameraThermo	213
21.8.1	HikvisionCameraThermo settings	213
21.9	HikvisionRecognitionOnBoard terminal	214
21.9.1	HikvisionRecognitionOnBoard settings	215
21.10	HikvisionTerminalThermo terminal	216
21.10.1	HikvisionTerminalThermo settings	217
21.11	LunaFast2NextGen	220
21.11.1	Configuring parameters for connecting to LunaFast2NextGen	220
21.12	LunaFast4A1	222
21.12.1	LunaFast4A1 settings	223
21.13	Panda	226
21.13.1	Panda settings	226
21.14	R20Face	227
21.14.1	R20Face settings	228

21.15 UniUbi terminal	229
21.15.1 UniUbi settings	230
21.16 VKVision02	231
21.16.1 VKVision02 settings	232
22 Pipelines	233
22.1 Apacs2FA	233
22.2 CreateBastionEvent	234
22.3 Custom2FA	235
22.4 LunaEventListener	236
22.5 MatchByPhoto	237
22.6 MatchByPhotoInCbsAlpha	238
22.7 MatchInformerWebHook	239
22.8 MatchInformerWebSocket	240
22.9 SendCardToR20Face	241
22.10 SendCarsToLaurent	242
22.11 SendCarsToSigur	242
22.12 SendThermalEventToLuna	243
22.13 SendToBars	244
22.14 SendToController	244
22.15 SendToDevice	245
22.16 SendToGrgFaster	246
22.17 SendToLuna	246
22.18 SendToParsec	247
22.19 SendToSalto	248
22.20 SendToSigur	249
22.21 Strazh2FA	249

1. Glossary

Term	Description
Liveness	A software method to confirm the vitality of a person by one or several images in order to prevent spoofing attacks
LUNA ACE	Biometric terminal VisionLabs LUNA ACE. For more details, see the device documentation.
LUNA PLATFORM	VisionLabs automated facial recognition system designed to process, collect, analyze, store, and compare biometric data obtained from facial images. For more details, see the system documentation.
LUNA CARS	A system designed for detection, tracking, recognition of vehicle, and license plates attributes. For more details, see the system documentation
Database	An organized collection of data stored and accessed electronically. Databases are structured to facilitate the storage, retrieval, modification, and deletion of data in conjunction with various data-processing operations. A database management system extracts information from the database in response to queries
Identification	Search for the most suitable biometric template by comparing the vectors of face features with a list of similar biometric templates in the database (one to many)
Software	A program or set of programs used to control a computer
Physical access control system (PACS)	A set of hardware and software tools aimed at controlling the entrance and exit in order to ensure safety and regulate visits to a particular facility
Face recognition system (FRS)	In the context of the document, VisionLabs products. For example, VisionLabs Access Control Server
Event	An immutable object that contains information about one person. The event is generated using an external system
Biometric system	A system designed for biometric recognition of individuals based on their behavioral and biological characteristics
Commercial Biometric System(CBS)	An organization accredited by the Ministry of Digital Resources of the Russian Federation to work with biometrics in accordance with Federal Law 572, PP RF № 810, as a result of which it has the right to store Vectors and perform authentication by biometrics using them, as well as to provide authentication services to third parties (organizations)

Term	Description
Simple Object Access Protocol (SOAP)	Web protocol for ensuring interaction between services, implemented in WSDL.
Universally unique identifier (UUID)	The name of objects (lists, events, cameras, etc.) that systems generate independently as a unique name.
Web Services Description Language (WSDL)	an XML-based interface description language that is used for describing the functionality offered by a web service

2. Introduction

The document describes the purpose and functions of the user interface of the service **VisionLabs LUNA Access** version 2.19.0 (hereinafter — Access), and also contains hardware and software requirements.

Access is a set of software technical controls and management tools that allows you to implement the collaboration of VisionLabs products and various access control and management systems.

Access solves the following tasks:

- adding video transmission devices with frames that LP or CBS will work with;
- adding auxiliary devices for reading data from magnetic pass cards or obtaining data on a person's temperature;
- sending requests to add/change data to the LP;
- receiving identification events;
- sending requests from LUNA PLATFORM to the PACS software about identification events;
- logging of events about an unidentified person's attempt to pass through the turnstile.

Integrations using Access, LUNA CARS/LP/CBS and external devices allow you to solve the following tasks:

- access control;
- improvement of accessibility and throughput of checkpoints;
- control of the time spent by employees, visitors, vehicles in the secure area;
- protection against unauthorized access attempts using Liveness technology.

The number of connected cameras, terminals, and turnstiles can be any and depends on the requirements for the deployed system but is limited by the license for VisionLabs products and the ACS capabilities.

Depending on the chosen solution, access control can be applied using face recognition or a magnetic pass card.

3. System requirements

Software Requirements (Table 1).

Table 1. Software requirements

Resource	Recommended
Processor (CPU)	Intel or AMD 64-bit, 2-core, 2.0GHz
Random access memory (RAM)	2 GB
Monitor resolution	1600x1200 px

Workstation hardware requirements (Table 2).

Table 2. Hardware Recommendations

Resource	Recommended
Web Browser	Google Chrome (version 117.0 and higher);
	Microsoft Edge (version 117.0 and higher);
	Mozilla Firefox (version 117.0 and higher);
	Safari.
Internet connection	A stable Internet connection with a data transfer rate of at least 1 Mbps from the user.

4. Supported components and versions

4.1. Supported components

Access allows you to add the following components to integrations (Table 3):

Table 3. Supported Components

Type	Supported	Notes
VisionLabs	LUNA PLATFORM 5 (LP5)	5.10 or later
	LUNA CARS	Installer v.2.10.1 and later. The request is made to LUNA CARS Analytics
	FaceStream	5.1.6 and later
ACS	APACS, Sigur, Bastion, Bolid, Parsec, PercoWeb, Strazh, RusGuard, Gate, Salto and Bars-X	The connection to ACS Bars-X takes place via a pipeline without service.
Devices	VisionLabs Terminal: LUNA ACE, LUNA Fast 4A1, LUNA Fast 8A1, LunaFast2NextGen	Terminal 8A1 is connected via device settings 4A1.
	Beware	Terminals: TFR80-210T1Q, TFR80-210
	BioSmart	Terminal: Quasar
	Dahua	Camera: Camera
		Thermal imager: Thermo
	Fortuna	Thermal imager: F315
	GrgFaster	Terminal: SV-M082f-C2
	Hikvision	Camera: DS-2CD3126G2-IS
		Thermal imaging terminals: DS-K1TA70MI-T, DS-K1T671TM-3XF, DS-K5671-3XF/ZU
		Terminals: DS-K1T341AMF, DS-K1T341AM, DS-K1T680D-E1
	Sunell (Panda)	Thermal imager: SN-T5/13, SN-F22-13
	Uni-Ubi	Thermal imager terminal: Uface 8-C temp, Uface 8T - temp
	Hi-Tech Security	Terminal: VK-Vision-02
	R20Face	Terminal: R20-Face-T8

4.2. Services version matching

Access supports the following services and firmware versions (Table 4):

Table 4. Supported external services

Name in Access	Original Name	Version
Apacs	APACS 3000	8.3.1.0 update 18
Bastion	Elsys Bastion-2 / 3	2.1.11.2337 and newer
Bolid	Bolid-Orion Pro	1.20.3 (build 11940)
	Orion Pro Integration Module	1.4, 1.5.1
CbsMts	KBS MTS	-
CbsAlpha	CBS Alpha	-
CbsVtb	KBS VTB	-
CbsAkbars	KBS Ak Bars	-
LunaStreams	VisionLabs FaceStream	from 5.1.6 and newer
Gate	Gate	1.22.95
Luna	VisionLabs LUNA PLATFORM	5.10 and newer
LunaAceConverter	LUNA ACE	1.2.23
LunaCars	VisionLabs LUNA CARS	
	LUNA CARS Installer	v.2.10.1 and newer
	LUNA CARS API	v.4.0.15 and newer
	LUNA CARS Stream	v.3.0.20 and newer
	LUNA CARS Analytics backend	v.4.0.8 and newer
	LUNA CARS Analytics frontend	v.2.0.61 and newer
	Parsec (ParsecNet3)	3.11.629 39 and newer
PercoWeb	PERCo-Web 2.0	4.30
Rusguard	RusGuard	3.3.1
Salto	Salto	6.6.3.0
		Package 6.6.3.94
		Service 4.23.3.595
Sigur	ACS Sigur	1.6.3.18.s and newer
SigurThroughDatabase	ACS Sigur	1.6.3.18.s and newer

Name in Access	Original Name	Version
Strazh	Rubezh Strazh	1.2.211201.648
CryptoPro	cryptopro-service	1.4.1 and newer

4.3. Device version matching

Access supports the following devices and firmware versions (Table 5):

Table 5. Supported devices

Device	Model	Version
Beward	TFR80-210T1Q, TFR80-210	1.2.13.0, 2.1.6.0
BioSmart	BioSmart Quasar	2.3.0.46
DahuaThermo	-	2.631.0000000.31.T
Fortuna315	-	Camera: V4.02.00, Thermal imager: 2.20.0.0.R26130.alpha8, Hardware versions: V1.0, Versions of the algorithm: smart2.0.0-06-2020.06.17.16:06:42
GrgFaster	SV-M082f-C2	Firmware versions: 1.004.30.3bb324.R, Hardware version: 1.0.0
HikvisionCamera	DS-2CD3126G2-IS	V5.5.134 build 200430
HikvisionCameraTherm	DS-2TD2617B-3/PA	V5.5.26 build 200317
HikvisionRecognition OnBoard	DS-K1T341AMF, DS-K1T341AM, DS-K1T680D-E1	V3.2.30 build 220210
HikvisionTerminalThern	DS-K1TA70MI-T, DS-K1T671TM-3XF, DS-K5671-3XF/ZU	V3.2.32 build 210525

Device	Model	Version
LunaFast4A1	DS-K1T680D-E1, DS-K1T341AMF, DS-K1T341CMF, DS-K1T341AM, VL LUNA FAST 4A1, VL LUNA FAST 8A1, 671, DS-K1T671M, ACT-T1341M, DS-K1T680DF-E1, DS-K5671-ZU	V3.2.30 build 210415, V3.2.30 build 210525, V3.2.30 build 210526, V3.2.30 build 210812, V3.2.30 build 211025, V3.2.30 build 220607, V3.2.30 build 220803, V3.2.30 build 221027, V3.2.33 build 210816, V3.2.35 build 220415, V3.2.35 build 220817, V3.3.40 build 250106
Panda	SN-T5/13, SN-F22-13	v3.6.0825.1004.1.0.23.0.0, v3.6.0840.1004.1.45.1.0.2
R20Face	R20-Face-T8	GD-V31.6222, GD-V32.7267
UniUbi	Uface 8-C temp, Uface 8T - temp, R20-Face-T8	GD-V30.7219, GD-V32.7247, GD-V32.7267
VKVision02	VANCOR VK VISION 02	v2156

4.4. Supported versions of controllers and converters

Access supports the following controllers, converters, and their firmware versions (Table 6):

Table 6. Supported controllers and converters

Model	Version
Controllers	
Gate 8000 Ethernet	8216/003 (4.06)
SigurController E900U	37
Strazh STR20-IP	1.2.211201.648
Strazh Rubezh STR20-IP	0.21.1
Strazh Rubezh STR1-AP	0.33.2
Bolid C2000-Ethernet	2.60
Bolid C2000-2	2.20, 2.50
RusGuard ACS-103-CE	6.20

Model	Version
RusGuard ACS-102-CE V2.0	7.46
Parsec NC-8000	3.8
Parsec NC-60-K	1.4
Bastion Elsys MB-NET	2.13
Bastion Elsys MB-light	2.74
Bastion Elsys NG-800	4.11
PercoController CTL14	2.2.37
PercoController CT/L04.2	3.0.0.78; 3.0.0.79; 3.0.0.81
Laurent-2	L213
UCM-2A	6103
AAM-LAN-8W	Only for APACS.
APOLLO AAN-100/AAN-32S/AAN-32N	Only for APACS.
Converters	
Gate Ethernet/Wiegand	04.06, 1.3.2003
S4A Wiegand to TCP/IP	6005

5. Working with Access

5.1. Access Localization

Access supports to localizations:

- English;
- Russian.

Press button **pyc|eng** to change localization (Figure 1).

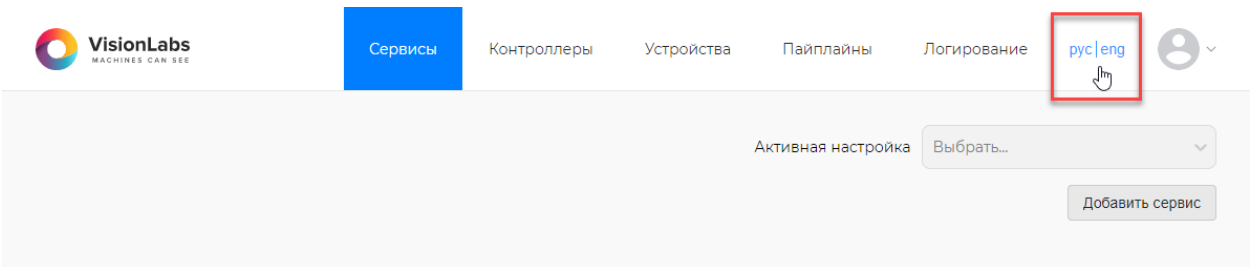


Figure 1. Interface localization

Access automatically refresh webpage and show english localization (Figure 2)

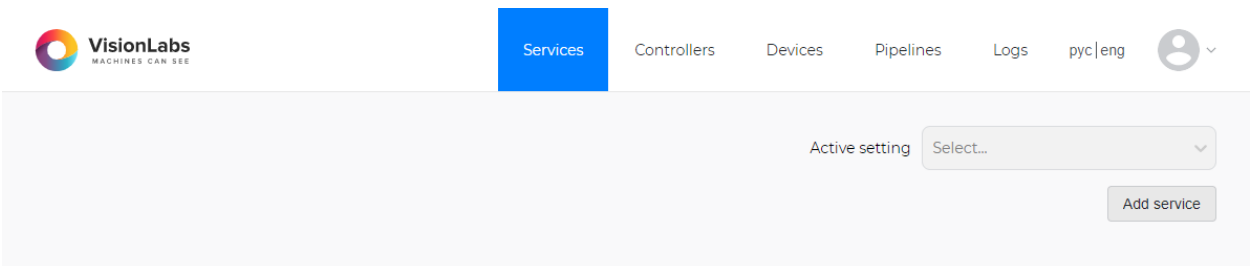


Figure 2. English interface

Для возврата к русскому интерфейсу нажмите на кнопку **pyc|eng** еще раз.

5.2. User roles and permissions

There is only one role available in Access — “administrator” (Table 7).

Table 7. List of available sections and permissions

Role	Section	Permission
Administrator	Services	Add/Edit/Delete services;
	Controllers	Add/Edit/Delete controllers;

Role	Section	Permission
	Devices	Add/Edit/Delete devices;
	Pipelines	Add/Edit/Delete pipeline;
	Logs	View logs;
		Export of logs

5.3. Adding an account

All user accounts are created by the administrator of the Access.

For a complete description of the user creation process, see the Administrator Manual.

5.4. Authorization in Access

The user accesses Access by logging into a web browser to the site.

You need to open a web browser and go to the server where Access was installed. Example of an address: `http://ip_address:9092/services`.

The link to enter the Access web interface must be requested from the administrator.

The first time you sign in to Access, the **Services** page is launched (Figure 3).

Unauthorized users are not allowed to view and create components and logs.

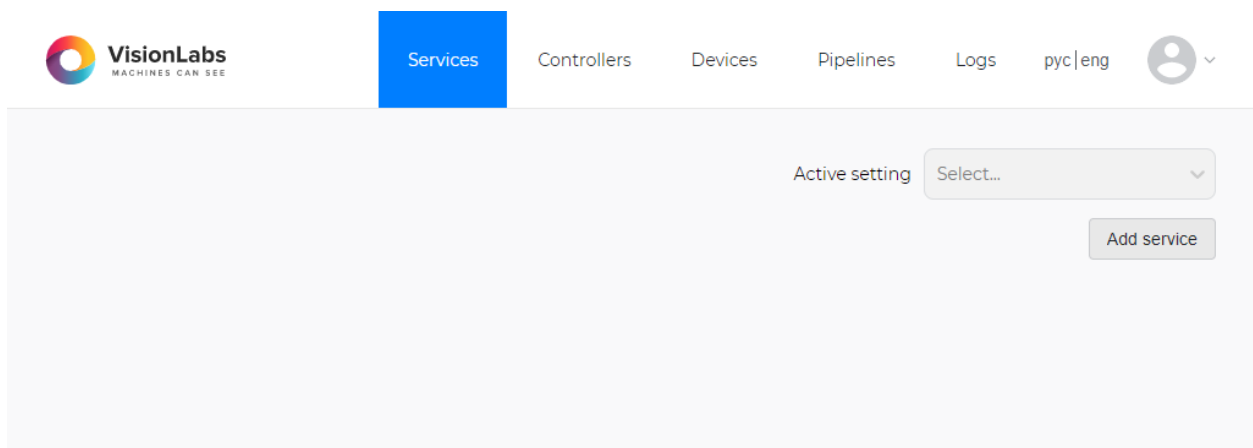


Figure 3. General view of the Access interface in a web browser

For authorization in the Access, click on the ▼ to the right of the user's avatar and click on the **Sign in** button (Figure 4).

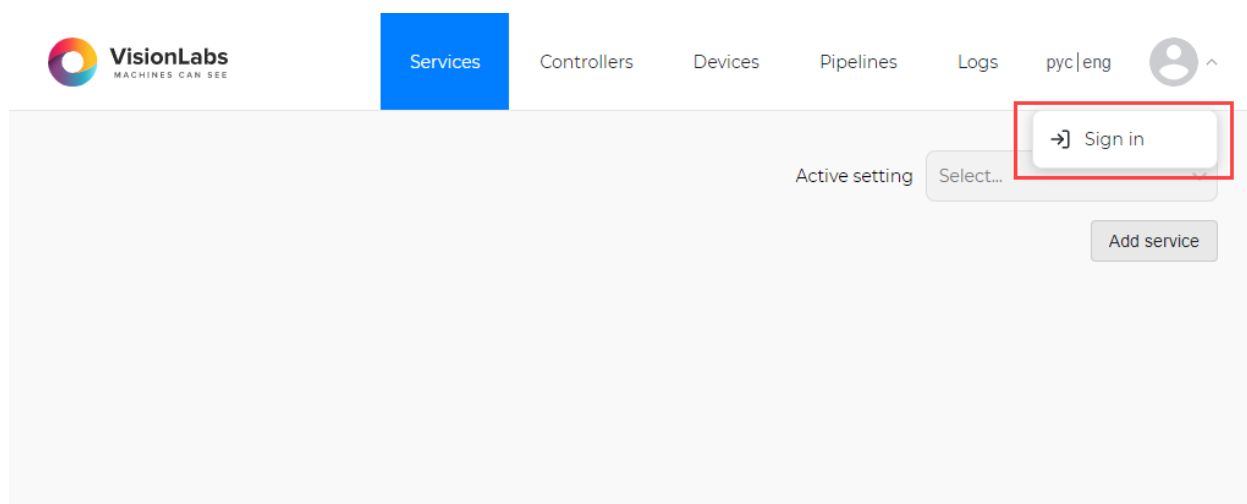


Figure 4. Login to user account

The authorization form will open (Figure 5).

To authorize in the Access, enter your credentials (login and password) in the appropriate fields and click the **Sign in** button.

The screenshot shows the authorization form. At the top is the VisionLabs logo. Below it is a light gray rounded rectangle containing two input fields: 'Login' and 'Password'. Below these fields is a blue button labeled 'Sign in'.

Figure 5. Authorization form

The login and password are requested from the administrator of the Access.

When entering the Access, the user is taken to the **Services** page (Figure 6), where he can configure and add the components of the Access.

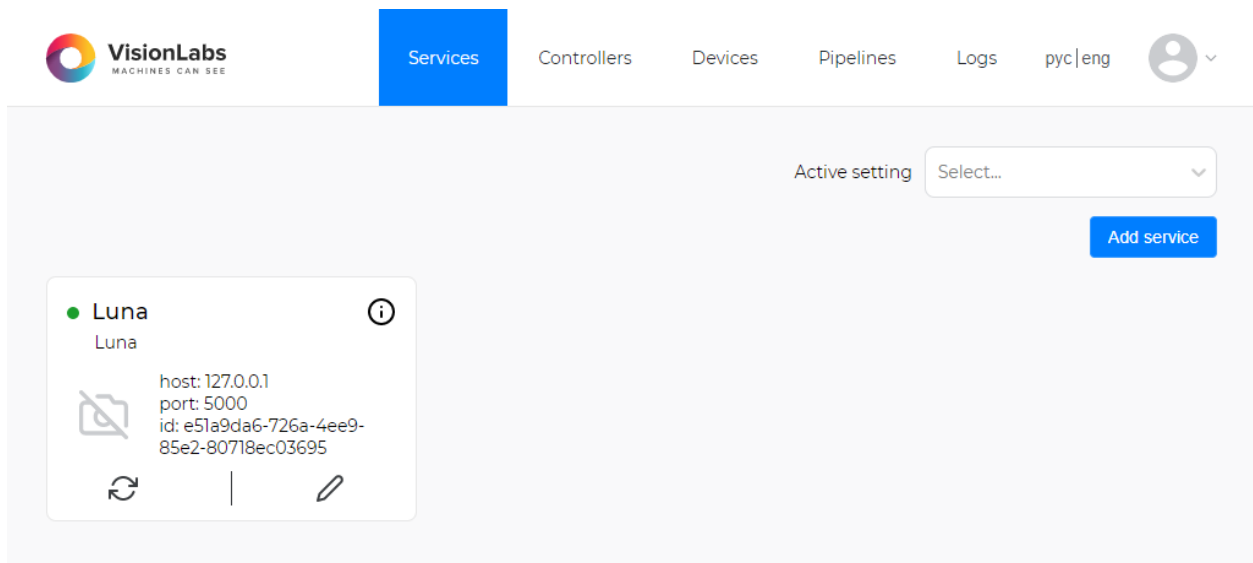


Figure 6. Screen of the page when the user is authorized in the Access

5.5. Signing out of the Access account

To sign out of the account, click on the to the right of the user's avatar and click on the **Logout** button (Figure 7).

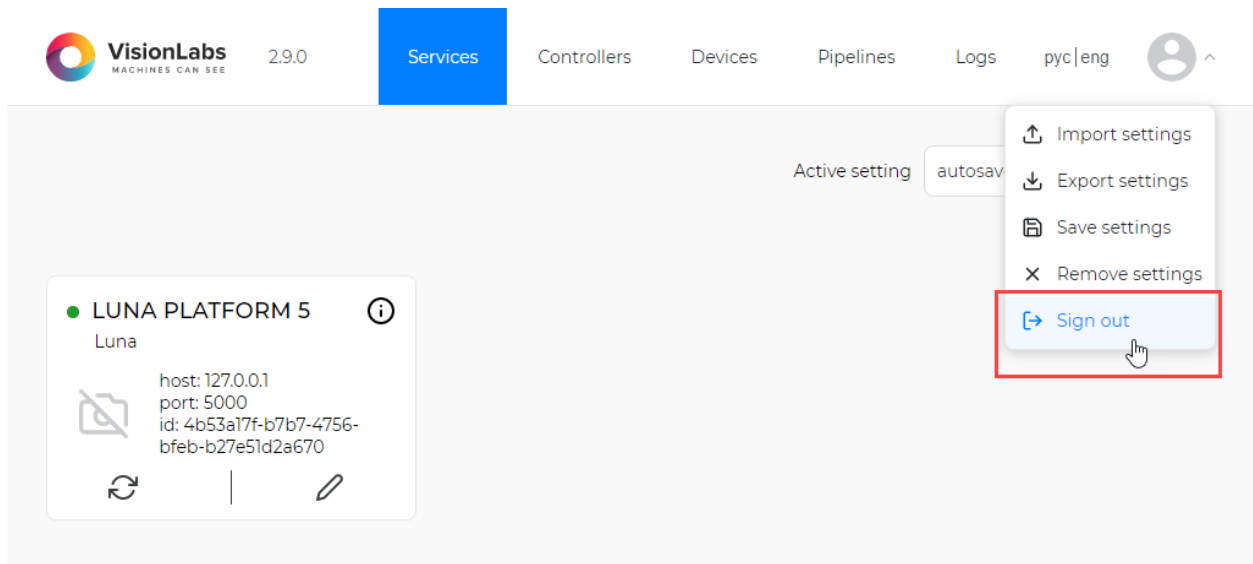


Figure 7. Signing out of a user account

After clicking on the **Logout** button, the user is redirected to a page with a general view of the Access interface in a web browser, where he will not be able to configure and add Access components.

5.6. Access sections

The user interface of the Access contains 5 sections in the main menu and 4 functions in the drop-down menu (Figure 8).

The main menu consists of the following sections:

- [Services](#) — view and create services;
- [Controllers](#) — viewing and creating controllers;
- [Devices](#) — view and create devices;
- [Pipelines](#) — view and create pipelines;
- [Logging](#) — viewing logs.

The drop-down menu consists of the following functions:

- [Import settings](#) — a function that allows you to import settings;
- [Export settings](#) is a function for exporting settings;
- [Reset settings](#) reset all settings;
- [Documentation](#) - open HTML documentation.

To expand the drop-down menu, click on the ▼ to the right of the user's avatar.

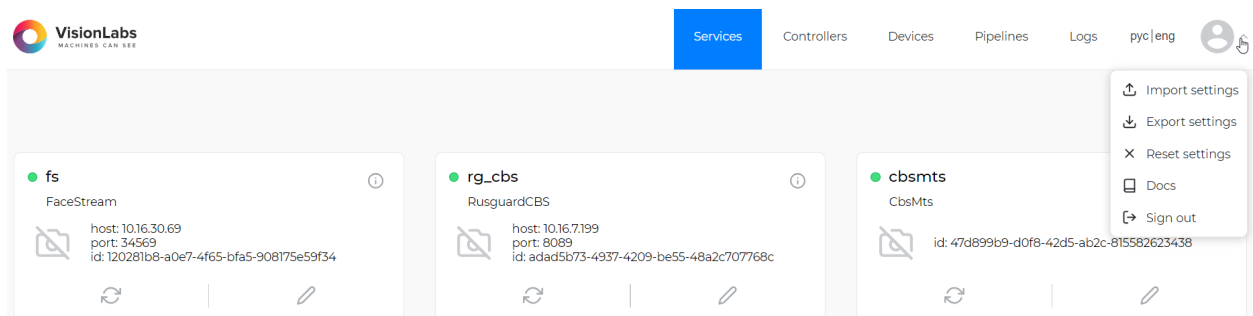


Figure 8. Menu sections and functions available to the administration

Components of the same type (Services, Controllers, Devices, or Pipelines) must have unique names (the `name` parameter).

Interaction with Access components occurs according to a common algorithm:

- [creation](#) of components
- [getting information](#) about the component
- [grouping](#) of components (available only for controllers and devices)
- [editing](#) a component
- [component restart](#)
- [removal](#) of a component

5.6.1. Creating a component

To create a new component, do the following:

1. click on the **Add component** button (Figure 9);

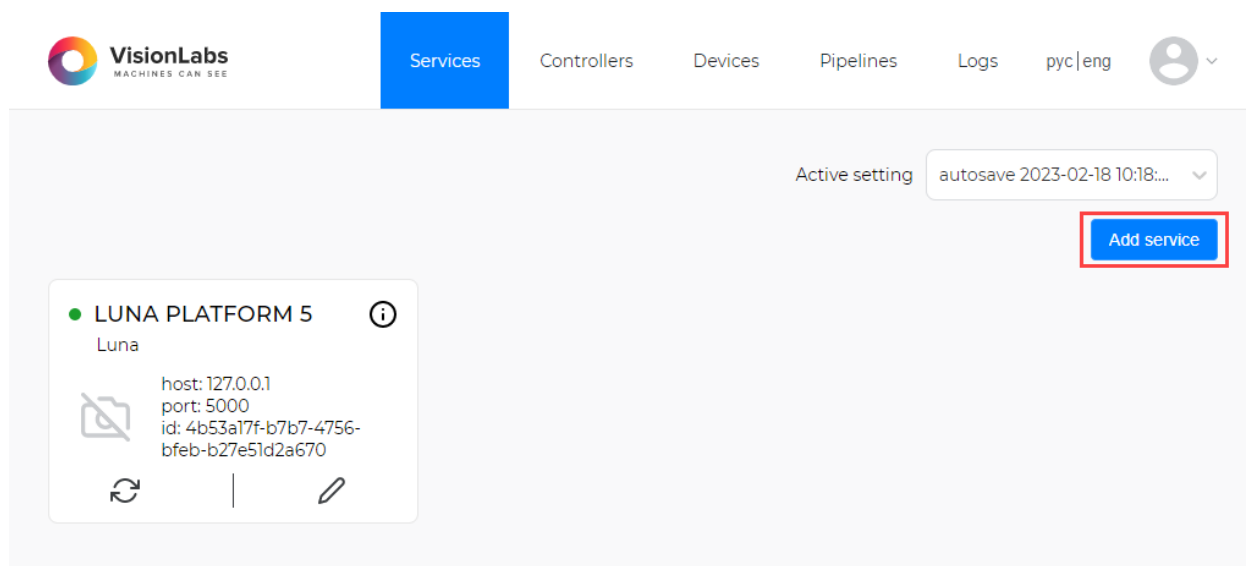


Figure 9. Creating a new component

2. a form for creating a component will open, in which you should select the type of component (Figure 10);

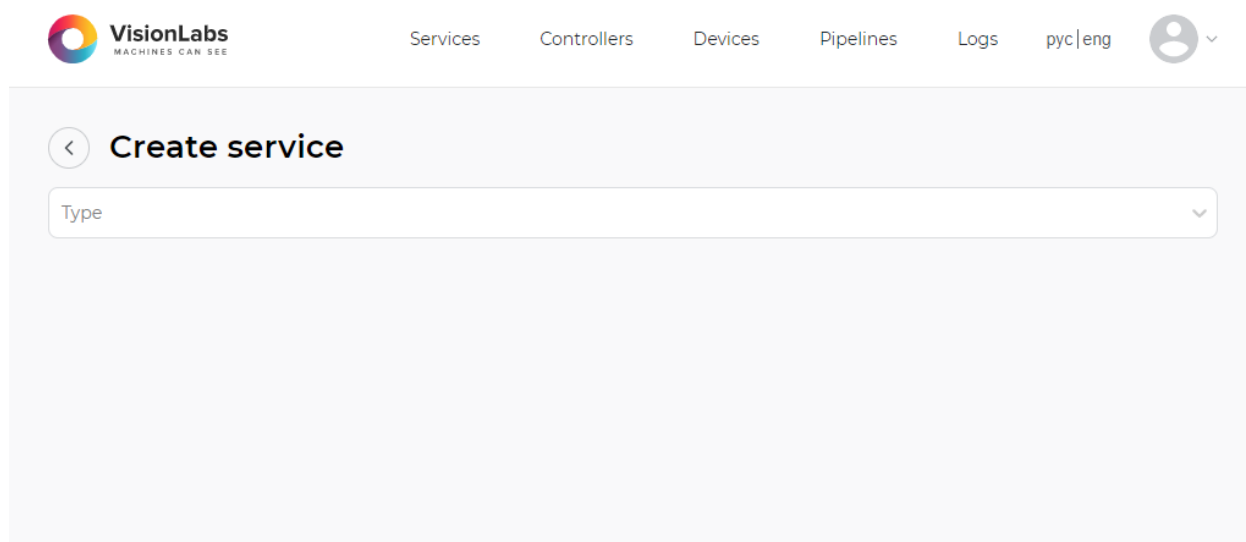


Figure 10. component creation form

3. to expand the drop-down menu, click on the ▼ on the right and select the required type of

component (Figure 11);

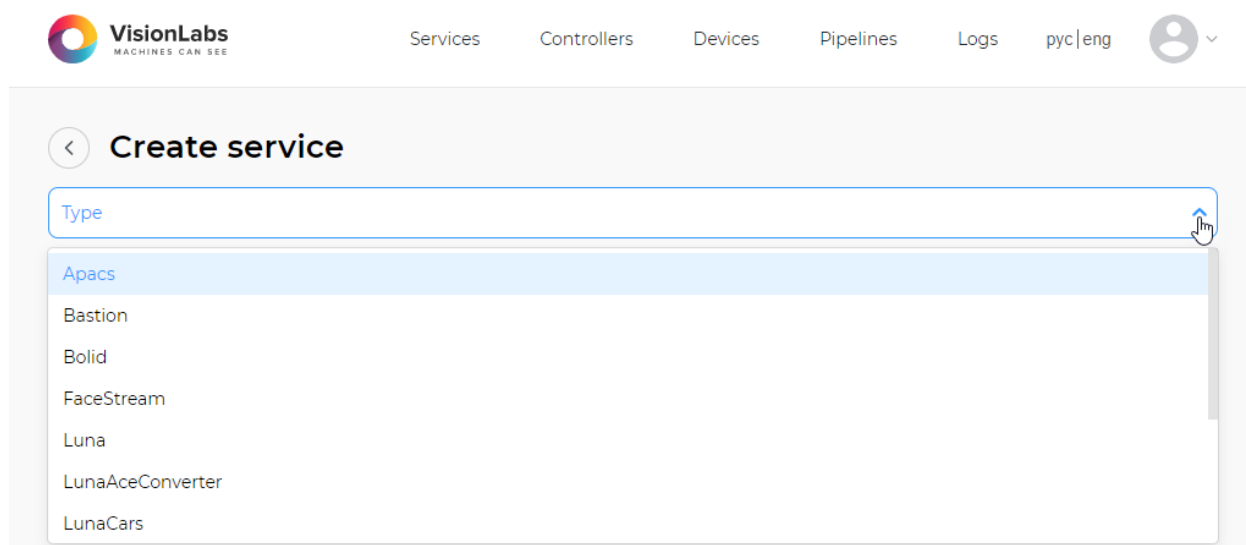
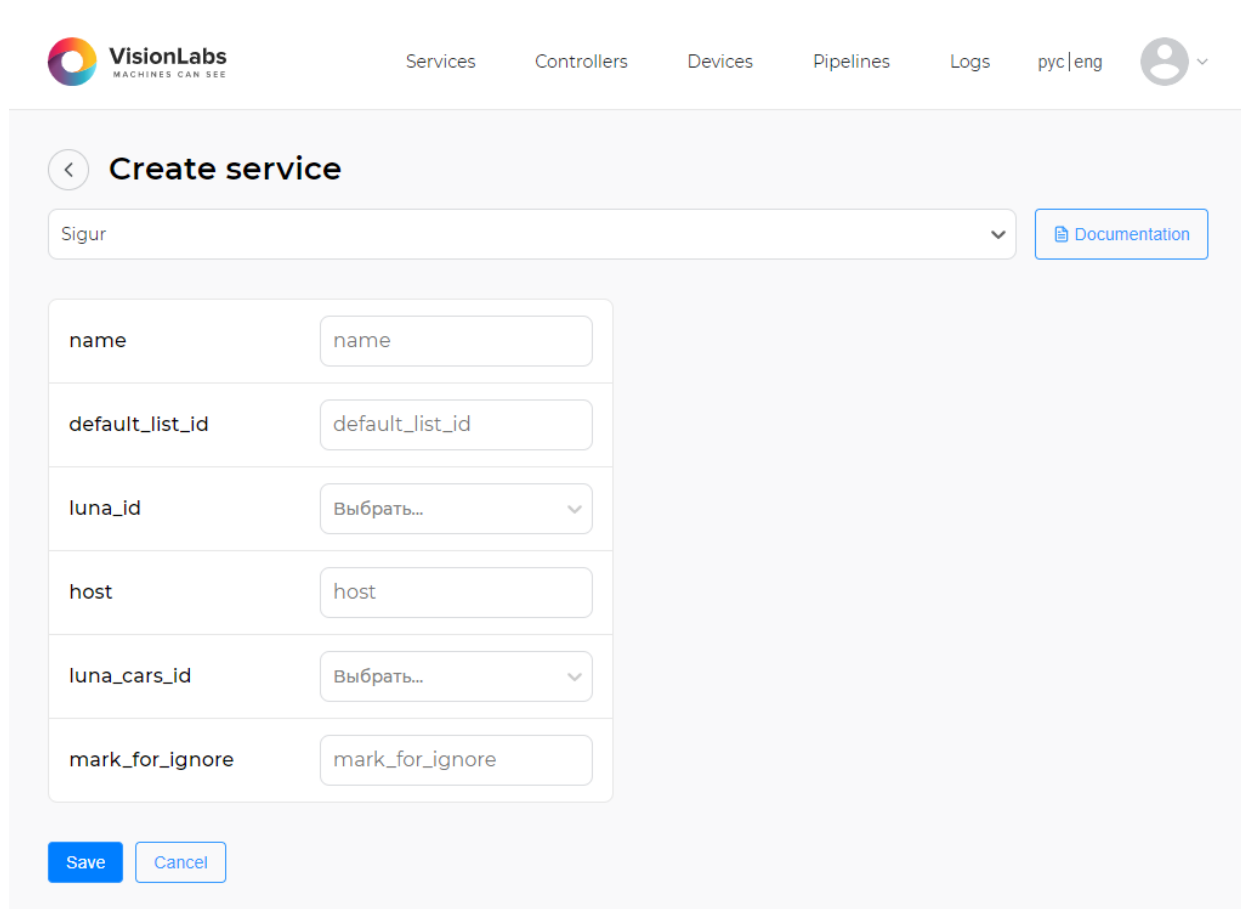


Figure 11. Selecting the type of component to create

4. a form will open for filling in the component settings, in which you should add the necessary Parameters (Figure 12);

The Parameters for configuring each component are different, see the [components](#) section.

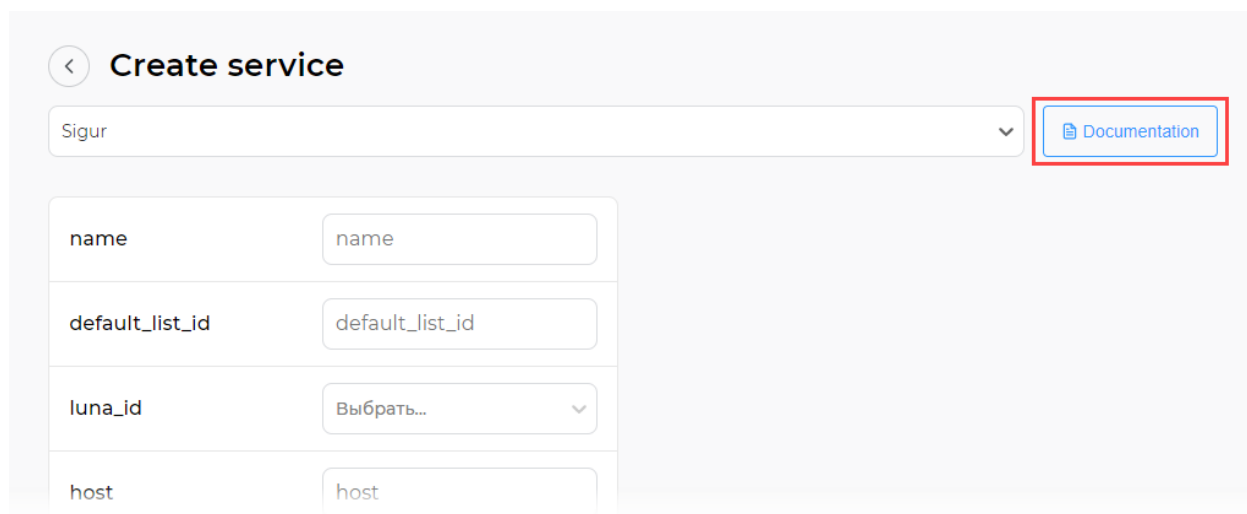


The screenshot shows the 'Create service' form in the VisionLabs LUNA Access interface. The top navigation bar includes the VisionLabs logo, the tagline 'MACHINES CAN SEE', and links for Services, Controllers, Devices, Pipelines, Logs, and a user profile icon labeled 'pyc | eng'. The form itself has a title 'Create service' with a back arrow. Below the title is a dropdown menu currently showing 'Sigur' and a 'Documentation' button. The main part of the form is a table with six rows, each containing a parameter name and a corresponding input field. The input fields for 'luna_id' and 'luna_cars_id' are dropdown menus with the text 'Выбрать...' (Select...). At the bottom of the form are 'Save' and 'Cancel' buttons.

Parameter	Input Field
name	name
default_list_id	default_list_id
luna_id	Выбрать...
host	host
luna_cars_id	Выбрать...
mark_for_ignore	mark_for_ignore

Figure 12. Form for filling in the component settings

to get information about configurable Parameters, click on the **Documentation** button in the upper left corner (Figure 13);



< **Create service**

Sigur ▼ [Documentation](#)

name	<input type="text" value="name"/>
default_list_id	<input type="text" value="default_list_id"/>
luna_id	<input type="text" value="Выбрать..."/> ▼
host	<input type="text" value="host"/>

Figure 13. Documentation for the created component

Pop-up window will display information describing the required Parameters for creating a component (Figure 14);

Sigur

The service is designed to interact with Sigur PACS

Supported Sigur software version: 1.1.1.9s

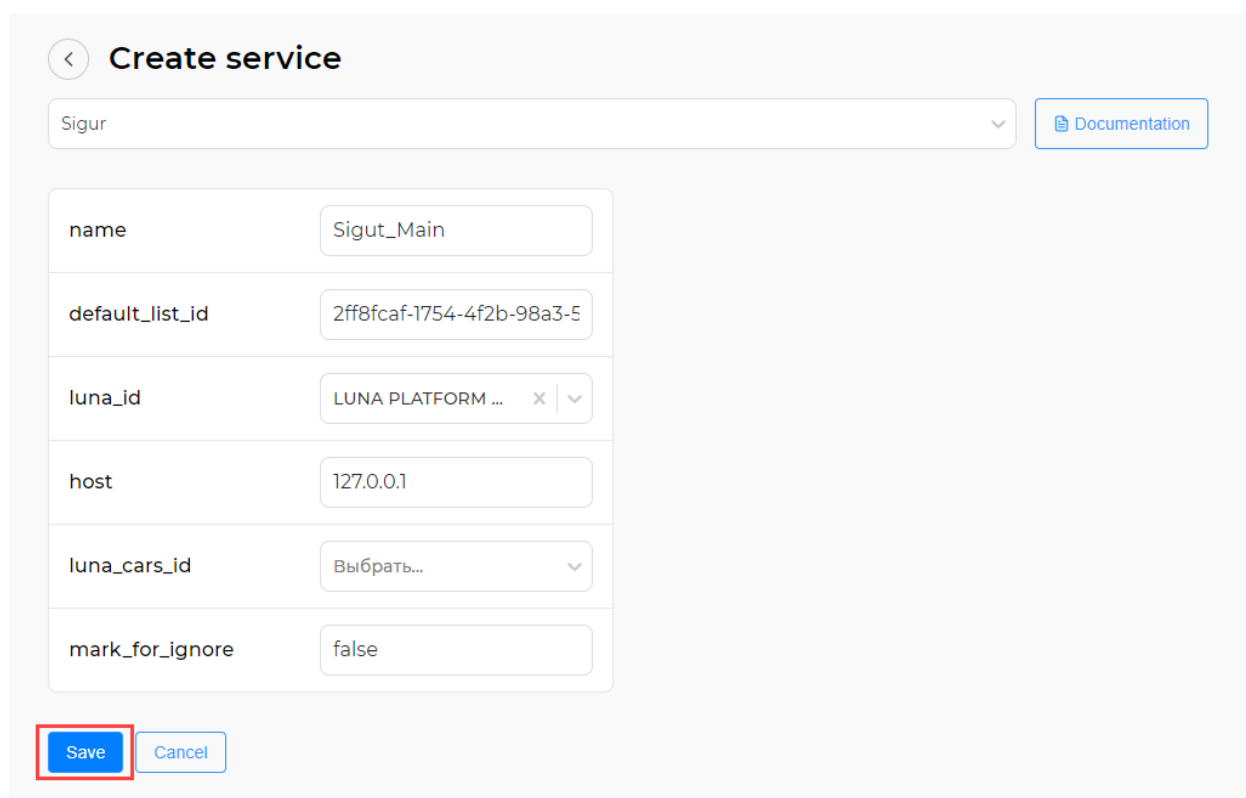
Sigur PACS synchronizes employees with the list in Luna and listens to events based on which it decides to open or not open the turnstile. These events are generated in VL Access by the SendToSigur pipeline.

The following settings are used when creating a new service:

- name: str - service name,
- default_list_id: str - ID of the Luna list that Sigur will synchronize employees with,
- luna_id: str - Luna service ID,
- host: str - device IP address,
- luna_cars_id: str - Luna Cars service ID,
- mark_for_ignore: str - when synchronizing, if this combination occurs in the body of the employee's name, then this name is ignored.

Figure 14. Pop-up window with the required component Parameters

5. after filling in the component Parameters, click on the **Save** button (Figure 15).



Create service

Sigur

Documentation

name	Sigut_Main
default_list_id	2ff8fcac-1754-4f2b-98a3-5
luna_id	LUNA PLATFORM ...
host	127.0.0.1
luna_cars_id	Выбрать...
mark_for_ignore	false

Save Cancel

Figure 15. Saving when creating a component

After successful creation of the component, the message **component has created successfully** will appear in the upper left corner of the screen (Figure 16).

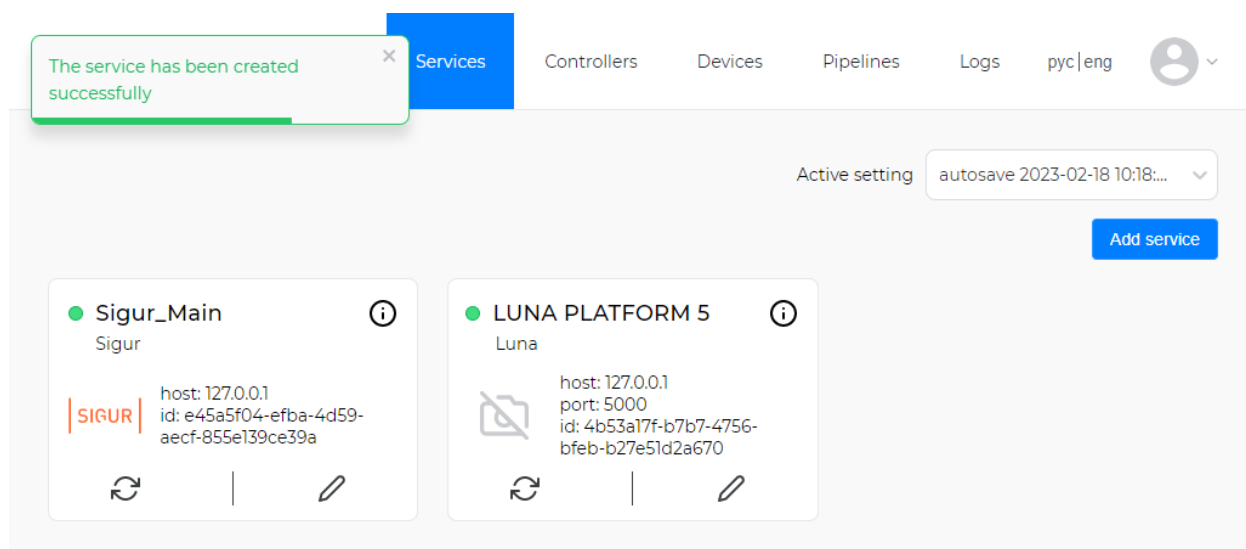


Figure 16. component creation confirmation

Upon successful creation, the new component should appear in the list of available components in the **components** section (Figure 17).

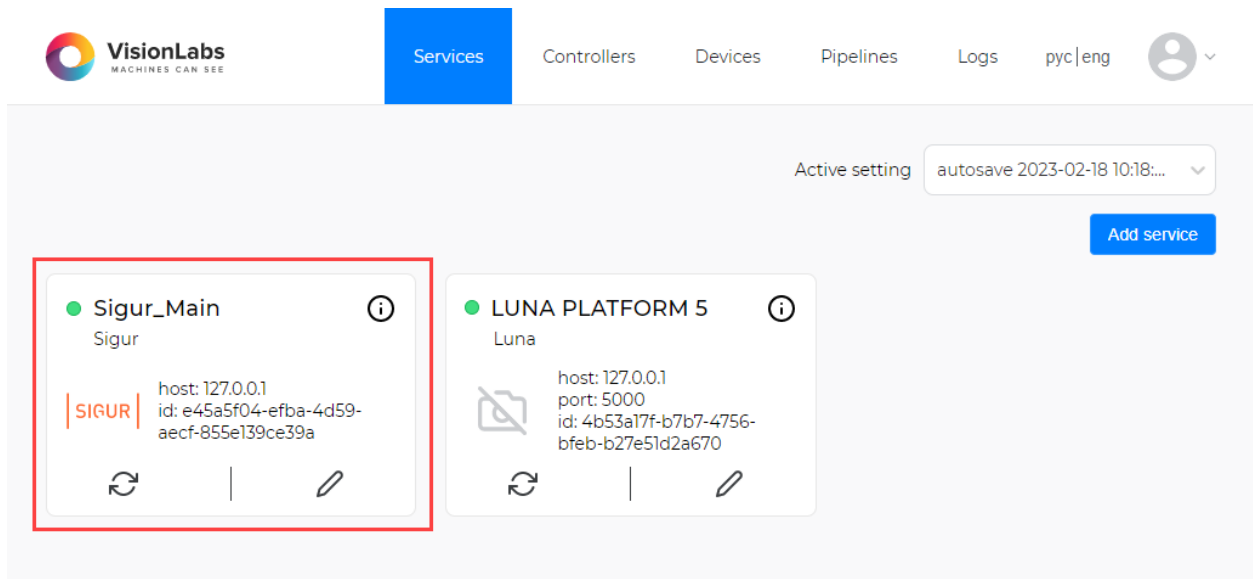


Figure 17. Displaying a new component

5.6.2. General information about the component

The available components are displayed on the general view page of the section.

General information about the component is located in the window with a description of the component (Figure 18).

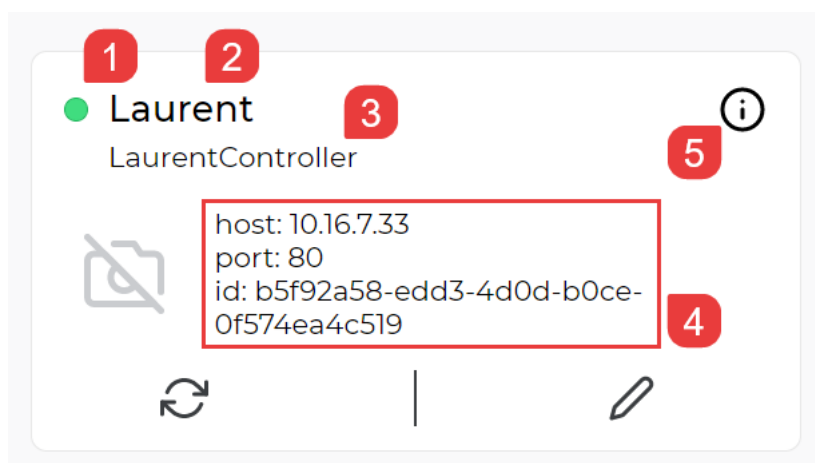


Figure 18. Main Parameters of the component

- 1 — status;
- 2 — name;

- 3 — type;
- 4 — general information ;
- 5 — additional information about Parameters.

For more information about the Parameters of the component, hover over ⓘ and information will appear in a pop-up window (Figure 19).

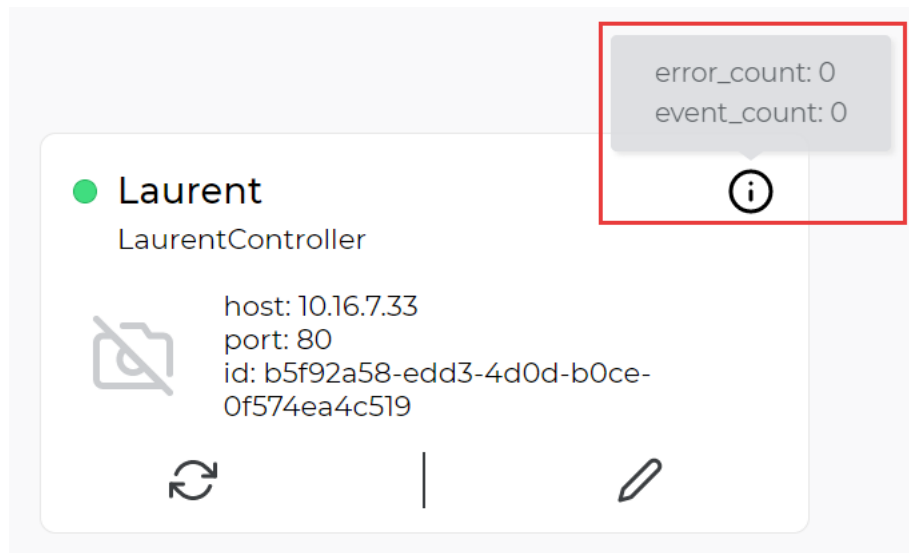


Figure 19. Additional information about component Parameters

5.6.3. Component grouping

When creating a component, the user can create/select a component group.

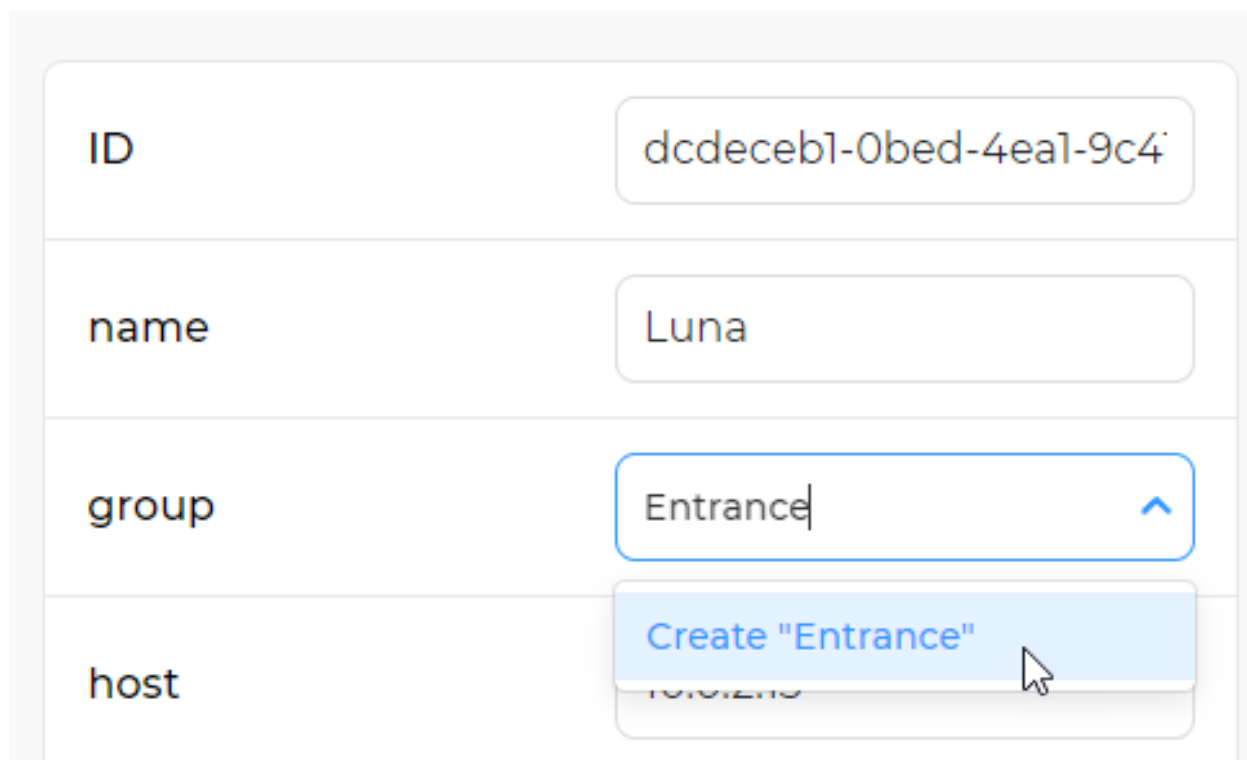
By default, components are assigned to the “No group” group.

Component grouping allows you to visually separate components by features that interest the user: location, type, manufacturer, etc.

To create a group in the component parameters editing window:

1. click on the group input field
2. specify the group name
3. click Create (Figure 20)
4. save the component changes.

It is not recommended to enter more than 30 characters.



ID: dcdeceb1-0bed-4ea1-9c41-4b184c32a6c9

name: Luna

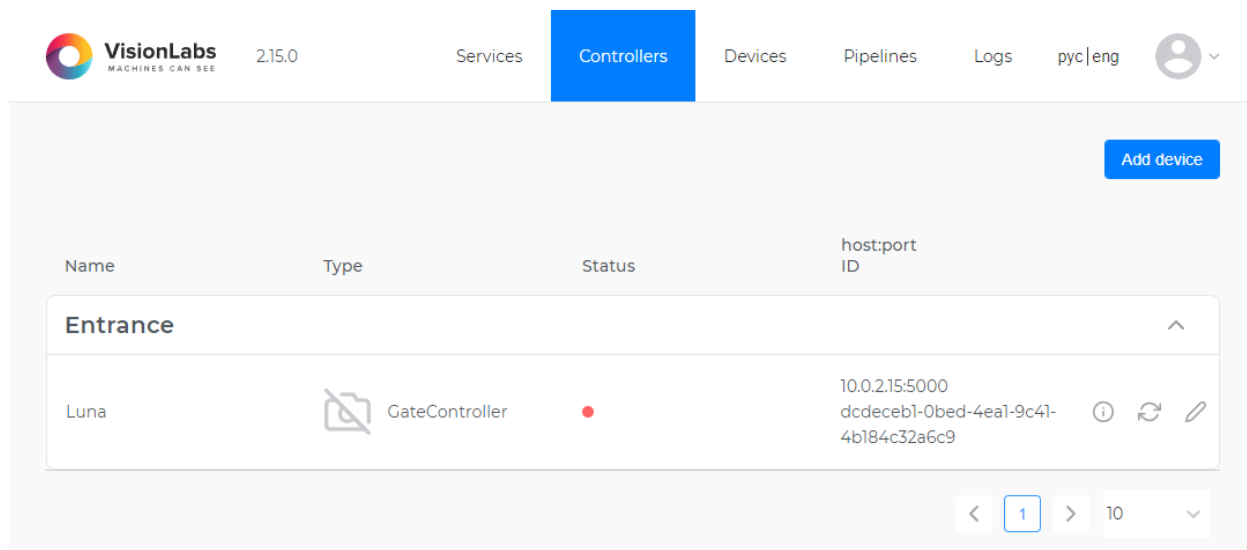
group: Entrance

host: 10.0.2.15

Create "Entrance"

Figure 20. Create group

The component is placed in the group (Figure 21).



VisionLabs 2.15.0 Services Controllers Devices Pipelines Logs pyc|eng

Add device

Name	Type	Status	host:port ID
Entrance	GateController	Red	10.0.2.15:5000 dcdeceb1-0bed-4ea1-9c41-4b184c32a6c9

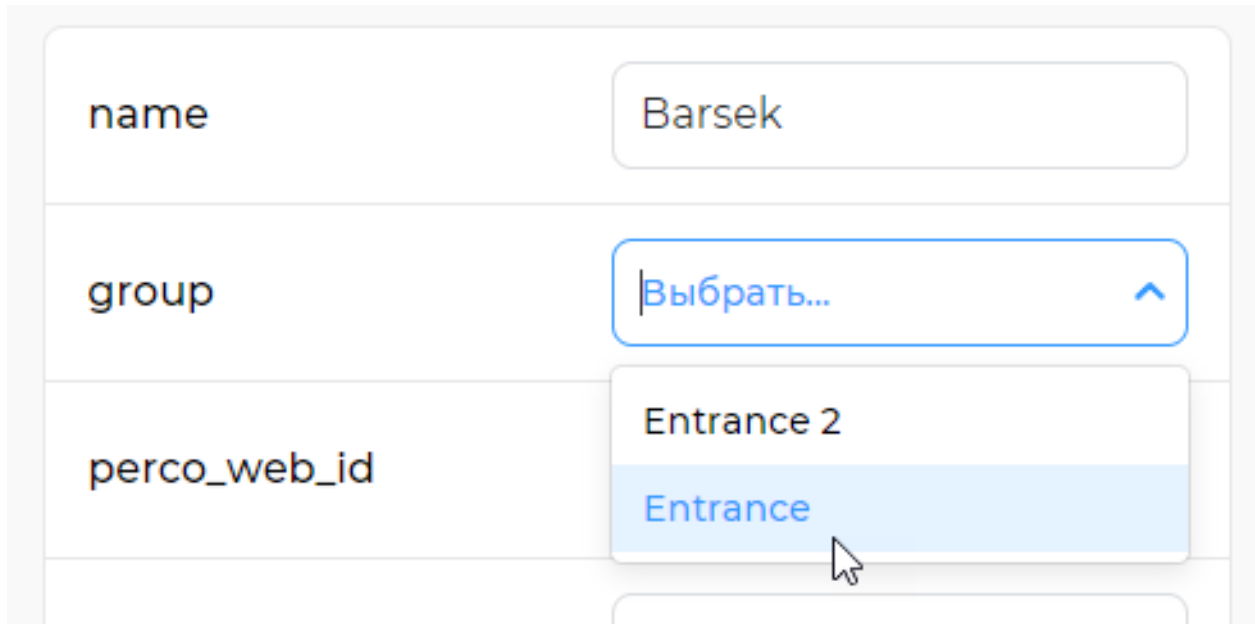
< 1 > 10

Figure 21. component group

To add a component to an existing group:

1. go to editing the component

2. select the desired group in the group drop-down list (Figure 22)
3. save the component changes.



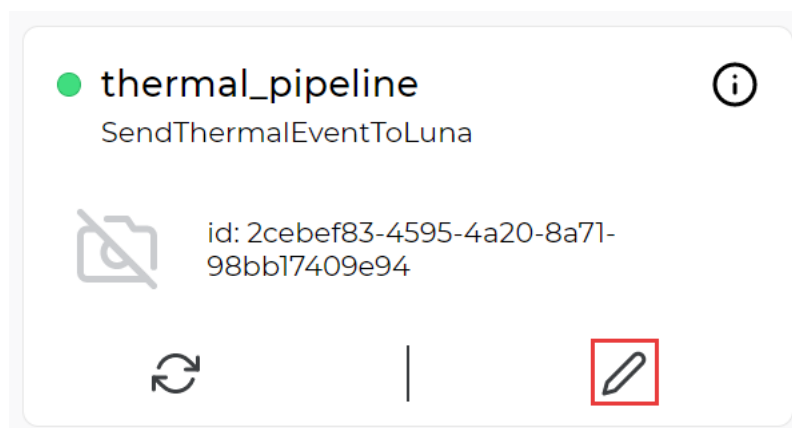
The screenshot shows a form with three fields: 'name' with the value 'Barsek', 'group' with a dropdown menu open, and 'perco_web_id'. The dropdown menu for 'group' shows two options: 'Entrance 2' and 'Entrance'. The 'Entrance' option is highlighted in blue, and a mouse cursor is pointing at it.

Figure 22. Add to group

5.6.4. Editing component

To edit the Parameters of component, do the following:

1. click on the  for the selected component (Figure 23);



The screenshot shows a component card for 'thermal_pipeline' with the function 'SendThermalEventToLuna'. It includes an ID: '2cebef83-4595-4a20-8a71-98bb17409e94'. At the bottom, there are three icons: a refresh icon, a vertical separator line, and a pencil icon (edit) which is highlighted with a red square.

Figure 23. Editing the component

2. a form for editing the component will open, in which you should make the necessary changes (Figure 24).

To get a description of the component parameters, you need to go to the corresponding section.

3. click on the **Save** button.

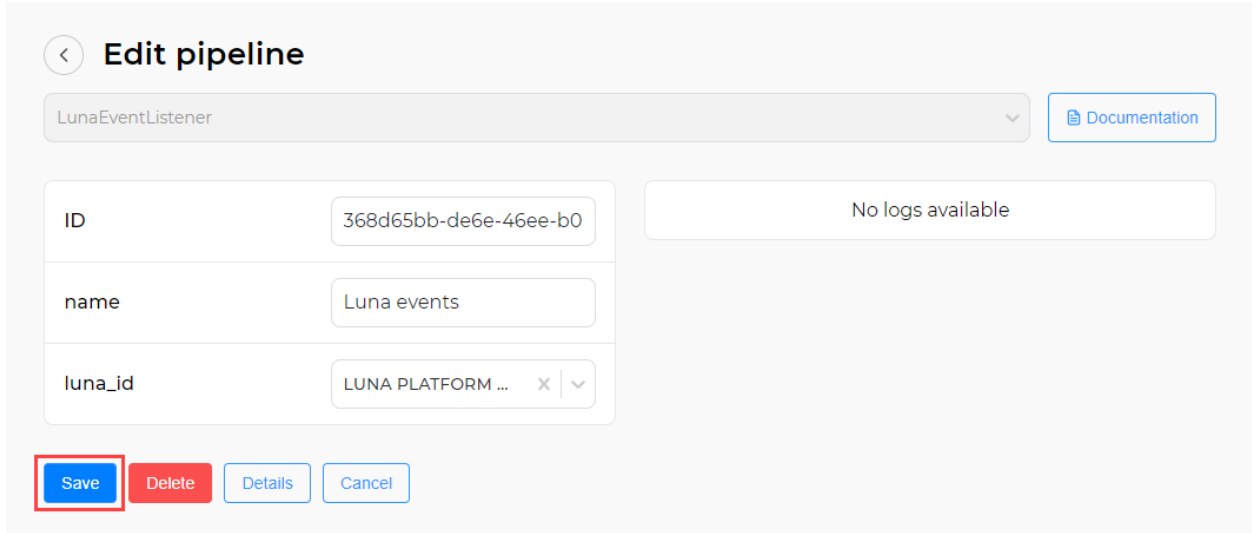


Figure 24. Component editing form

After successfully editing the component, the message **component has been updated** will appear in the upper left corner of the screen (Figure 25).

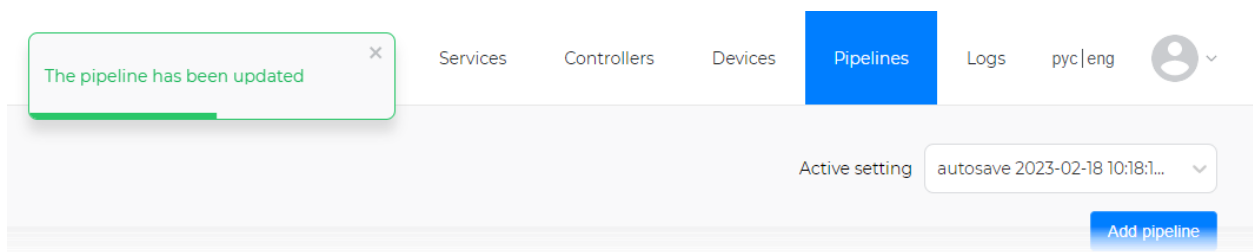


Figure 25. Component update confirmation

5.6.5. Restarting a component

To restart the component, click the  for the selected component (Figure 26).

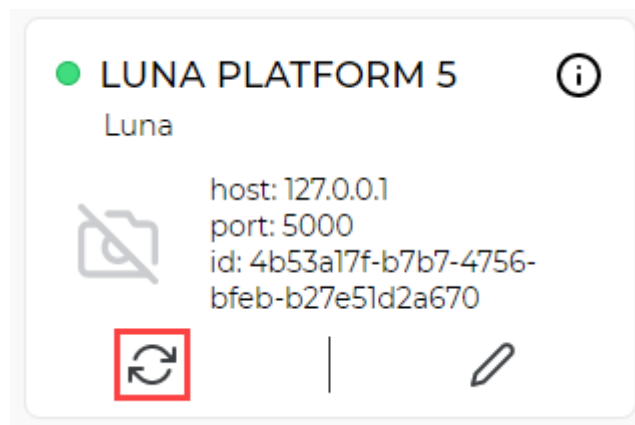


Figure 26. Restarting the component

After a successful restart of the component, the message **component has been restarted** will appear in the upper left corner of the screen (Figure 27).

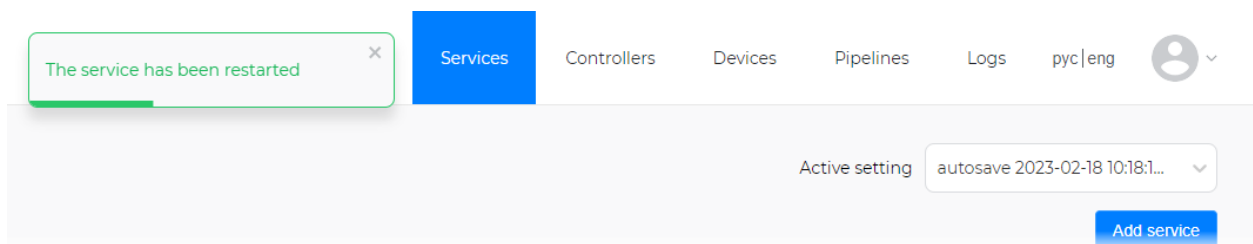


Figure 27. Component restart confirmation

5.6.6. Component removal

To remove an available component, do the following:

1. click on the  for the selected component (Figure 28);



Figure 28. Removing a component

2. a form for editing the component will open, in which you should click on the **Delete** button in the lower left corner (Figure 29).

< Edit service

Luna [Documentation](#)

ID	4b53a17f-b7b7-4756-bfeb
name	LUNA PLATFORM 5
host	127.0.0.1
port	5000
username	admin@test.ru
password	adminadmin
handler_id	2ff8caf-1754-4f2b-98a3-5
face_detection_threshold	0,5

[Save](#) [Delete](#) [Details](#) [Cancel](#)

18.02.2023, 10:18:52 INFO Component Luna: id: 4b53a17f-b7b7-4756-bfeb-b27e51d2a670 is successfully created in celery process.

Figure 29. Removing a component

After successfully deleting the component, the message **component has been removed** will appear

in the upper left corner of the screen (Figure 30).

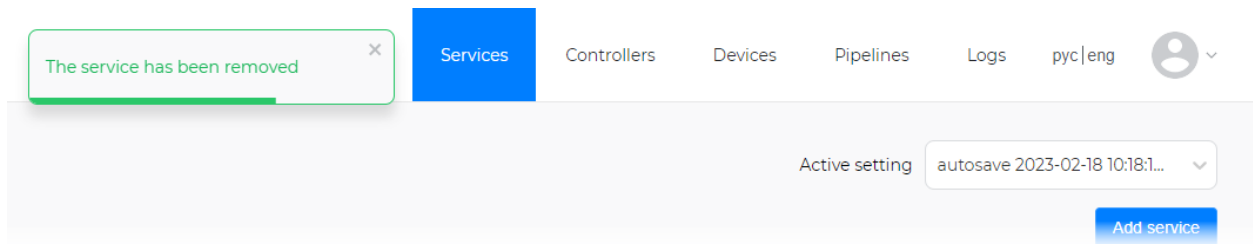


Figure 30. Component removal confirmation

Upon successful removal, the component will disappear from the list of available components in the **components** section.

Deleting a component also deletes its child components.

The child components are the corresponding controllers:

- Apacs service — ApacsController;
- PercoWEB service — PercoController;
- Salto service — SaltoController;
- Strazh service — StrazhController.

6. Logs

The **Logs** section is designed to display all system logs and search for the necessary logs in the history. Receiving and displaying new logs is performed with minimal delays in near real time.

Access stores logs for the last 7 days in an internal database, after which the logs are deleted.

The general view of the **Logs** section is shown below (Figure 31).

The screenshot shows the VisionLabs LUNA Access interface. At the top, there is a navigation bar with the VisionLabs logo and several menu items: Services, Controllers, Devices, Pipelines, Logs (highlighted in blue), and a user profile icon labeled 'rus|eng'. Below the navigation bar is a 'Filters' section with three main areas: 'Date from' and 'Date to' (each with a calendar icon and a close button), 'Logging levels' (with checkboxes for Debug, Error, Info, and Warning), and 'Order by' (a dropdown menu currently showing 'Create time from new to old'). Below the filters are three buttons: 'Filter' (blue), 'Reset' (orange), and 'Export' (blue). The main area displays a list of log entries. Each entry has a status icon (INFO or DEBUG), a timestamp, an IP address, and a message. To the right of each entry is a downward-pointing arrow icon. The log entries are as follows:

Status	Timestamp	IP Address	Message
INFO	15.02.2024, 12:08:02	10.16.7.152:51319	"POST /vl-access/webhook/device/0bf55b4d-eb54-4b10-8916-31d6d1094cd8/event/handle_event/ HTTP/1.1" 404
INFO	15.02.2024, 12:08:02	10.16.6.216:48875	"POST /vl-access/webhook/device/71e2167e-be58-4aaf-b173-0b0e706ad3a4/event/handle_event/ HTTP/1.1" 404
INFO	15.02.2024, 12:07:56	10.16.7.152:51288	"POST /vl-access/webhook/device/0bf55b4d-eb54-4b10-8916-31d6d1094cd8/event/handle_event/ HTTP/1.1" 404
DEBUG	15.02.2024, 12:08:01		Replication is in process. Counter: 16. Current employee: 3a363f68-36b5-4e2f-8f6d-f93b6dade6d7, Ткачев Сергей Сергеевич[0m
DEBUG	15.02.2024, 12:08:01		<PacsPerson: 3a363f68-36b5-4e2f-8f6d-f93b6dade6d7 Ткачев Сергей Сергеевич> does not need to be updated. He will be skipped.[0m
DEBUG	15.02.2024, 12:08:01		<PacsPerson: 479a10d0-1b40-4431-b7e5-cf4124e13205 Юнсовский Самат Абдыкадырович> does not need to be updated. He will be skipped.[0m

Figure 31. Logs section

In order to expand the detailed information about the target log, click on the arrow ▼ to the right of the required log (Figure 32).

VisionLabs
MACHINES CAN SEE

Services Controllers Devices Pipelines **Logs** pyc | eng

Filters

Time period

Logging levels
☐ Error
☐ Info
☐ Warning

Components
☐ Services
☐ Controllers
☐ Devices
☐ Pipelines

Filter **Reset** **Export**

19.02.2023, 11:20:56 INFO Component <LunaEventListener: 368d65bb-de6e-46ee-b0a9-c33ee9054c67> is successfully updated

19.02.2023, 11:06:05 ERROR Error for successful response

19.02.2023, 11:06:05 INFO {'code': 'LAN_EXP-1001', 'msg': 'The password is wrong, please check the correctness of the password', 'result': 1, 'success': False}

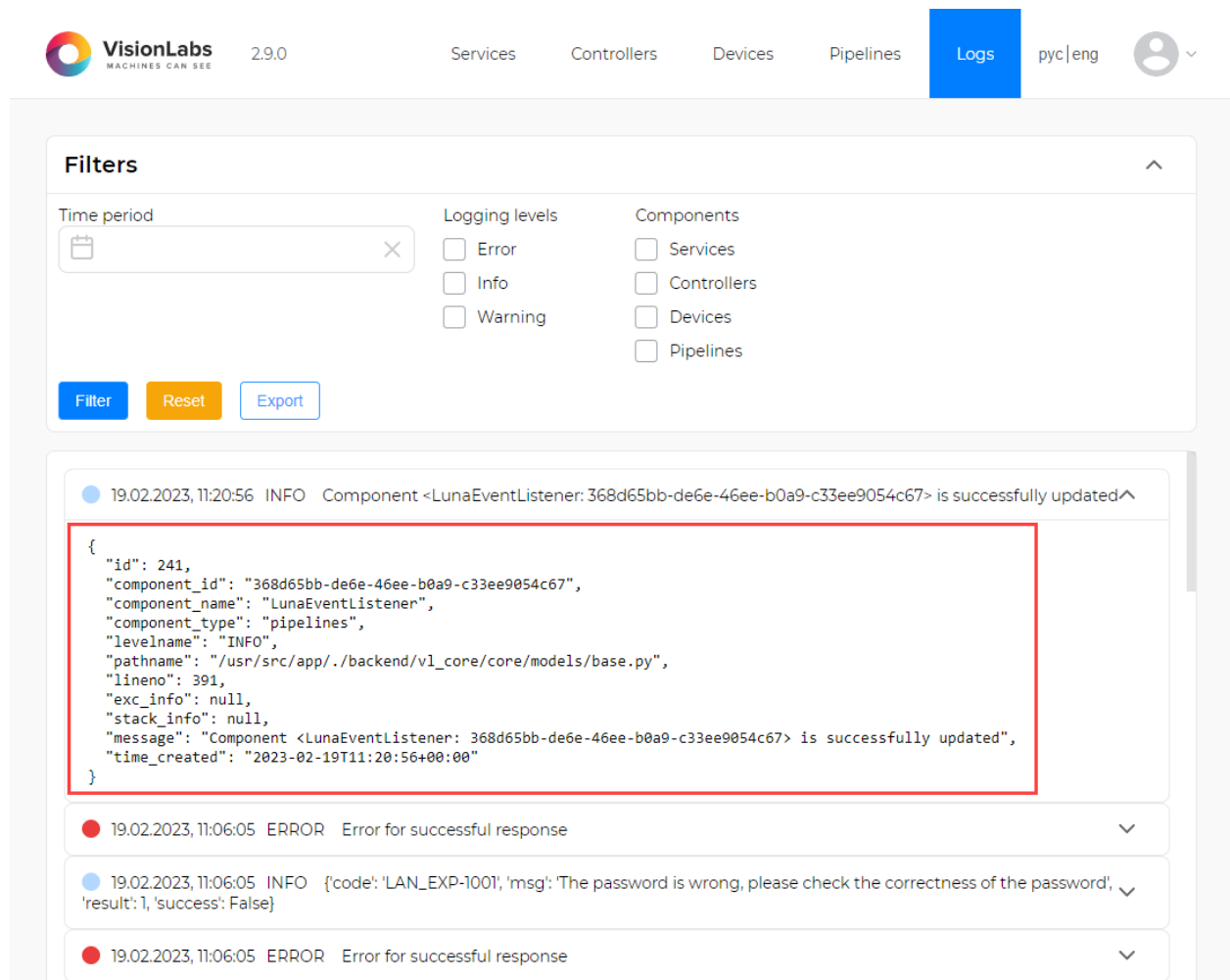
19.02.2023, 11:06:05 ERROR Error for successful response

19.02.2023, 11:06:05 INFO {'code': 'LAN_EXP-1001', 'msg': 'The password is wrong, please check the correctness of the password', 'result': 1, 'success': False}

19.02.2023, 11:06:05 ERROR Error for successful response

Figure 32. Detailed information on the logs

The target log opens with detailed information (Figure 33).



The screenshot shows the VisionLabs LUNA Access interface. At the top, there's a navigation bar with the VisionLabs logo, version 2.9.0, and links for Services, Controllers, Devices, Pipelines, and Logs (which is highlighted in blue). A user profile icon is on the right.

Below the navigation bar is a 'Filters' section with a 'Time period' dropdown, 'Logging levels' (Error, Info, Warning), and 'Components' (Services, Controllers, Devices, Pipelines). There are 'Filter', 'Reset', and 'Export' buttons.

The main area displays a list of log entries. The first entry is expanded, showing a detailed JSON log entry:

```
{
  "id": 241,
  "component_id": "368d65bb-de6e-46ee-b0a9-c33ee9054c67",
  "component_name": "LunaEventListener",
  "component_type": "pipelines",
  "levelname": "INFO",
  "pathname": "/usr/src/app/.backend/v1_core/core/models/base.py",
  "lineno": 391,
  "exc_info": null,
  "stack_info": null,
  "message": "Component <LunaEventListener: 368d65bb-de6e-46ee-b0a9-c33ee9054c67> is successfully updated",
  "time_created": "2023-02-19T11:20:56+00:00"
}
```

Below this, there are three more log entries:

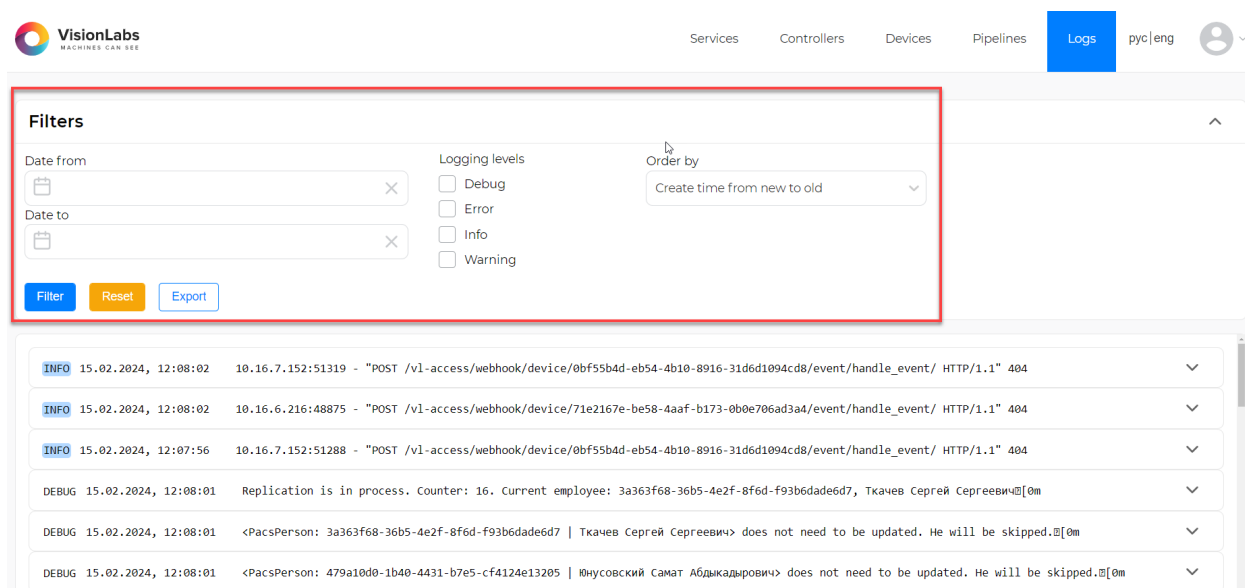
- 19.02.2023, 11:06:05 ERROR Error for successful response
- 19.02.2023, 11:06:05 INFO {'code': 'LAN_EXP-1001', 'msg': 'The password is wrong, please check the correctness of the password', 'result': 1, 'success': False}
- 19.02.2023, 11:06:05 ERROR Error for successful response

Figure 33. Detailed information on the logs

6.1. Logs filtering

The **Logs** section allows user to filter the latest logs in order to limit the display of logs on the screen (Figure 34).

Using filters, the user can quickly find the required log.

Figure 34. Filters in the **Logs** section

Filters available to the user in the **Logs** section:

- **Time period** — selection of time period for logging;
- **Logging levels** — selection of logging level (**Error**, **Info**, **Warning**);
- **Order by** — selection of logs sorting option (**From old to new**, **From new to old**).

The user needs to tick the required filter(s), and click the **Filter** button so that the settings are applied (Figure 35).

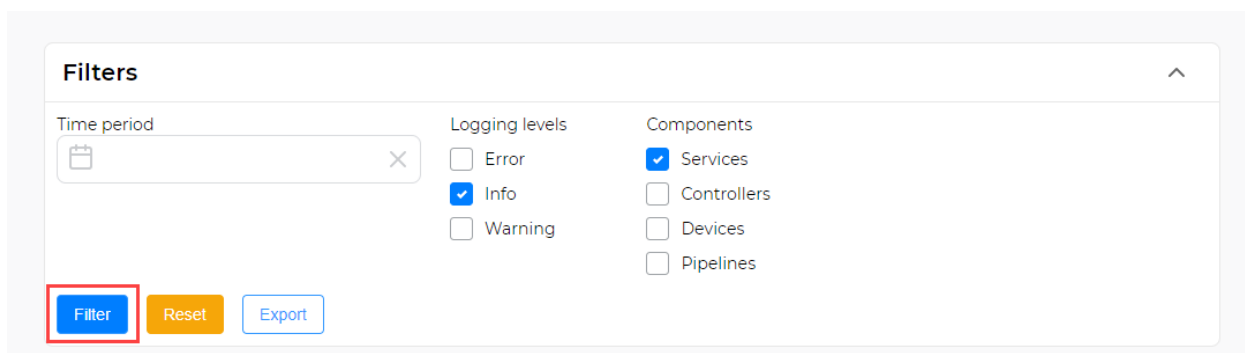


Figure 35. Filter selection

If the filters are applied successfully, information with logs should be displayed, considering the selected filters. (Figure 36).

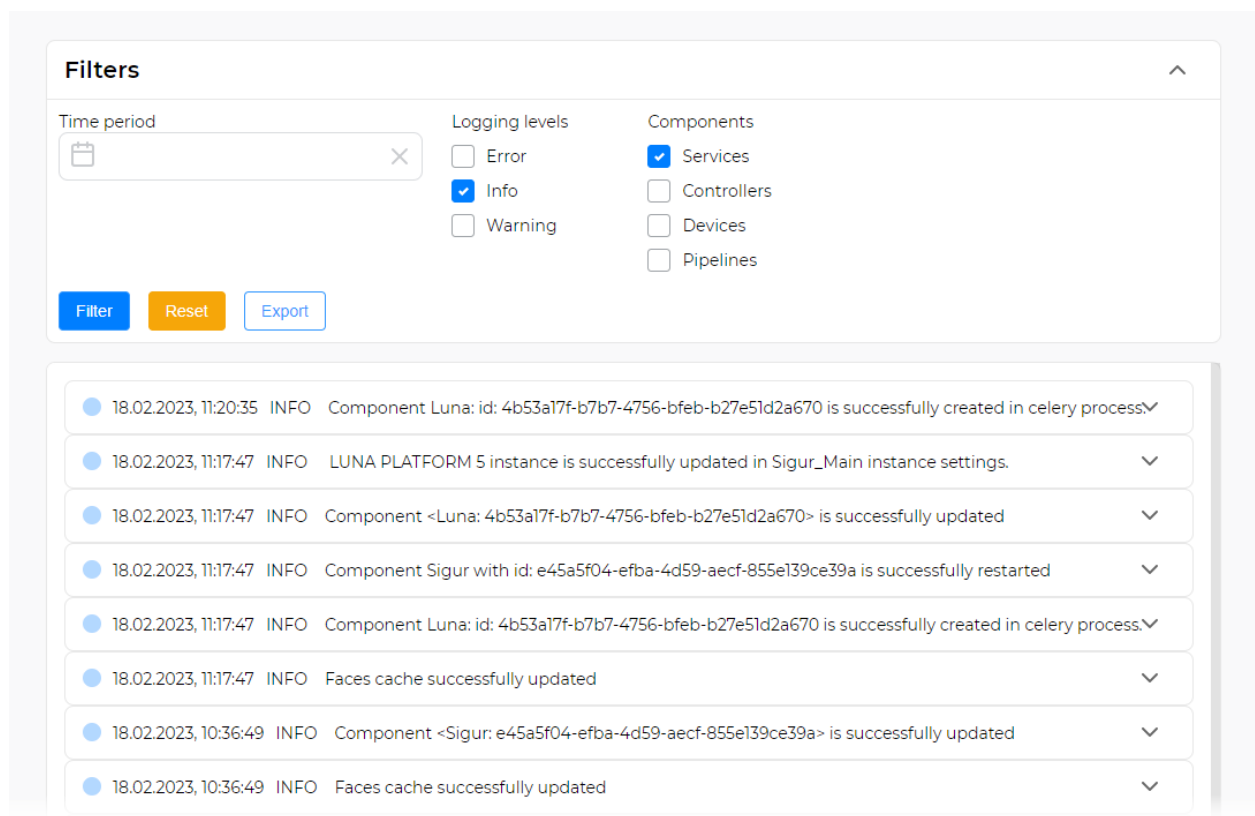


Figure 36. Displaying logs after filtering

To reset the selected filters, click the **Reset** button(Figure 37).

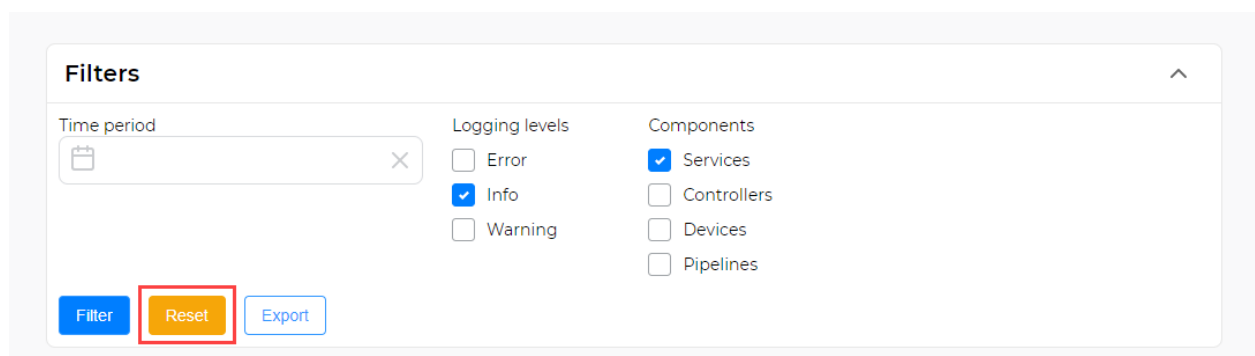


Figure 37. Resetting the filter settings

If necessary, you can export filtered logs. To do this, click the **Export** button (Figure 38). Logs are saved to the local computer in the txt format.

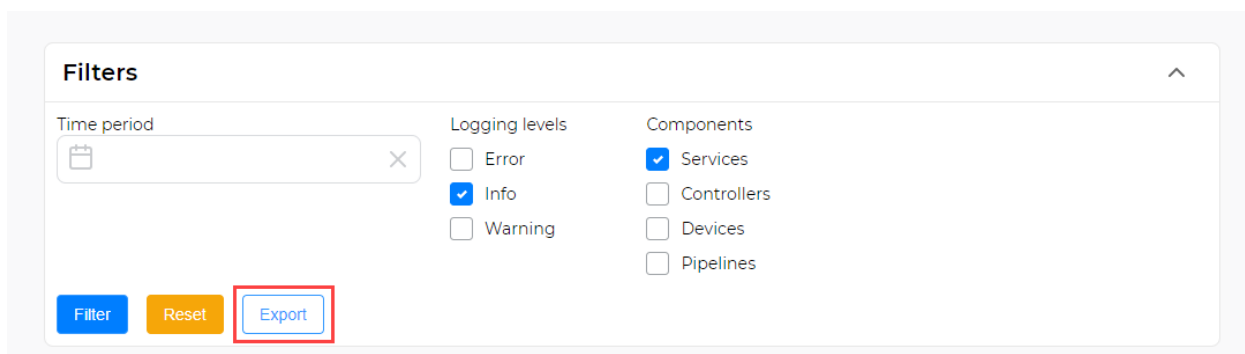



Figure 38. Log export

To collapse the Filters block, click on the arrow  in the upper right corner (Figure 39).

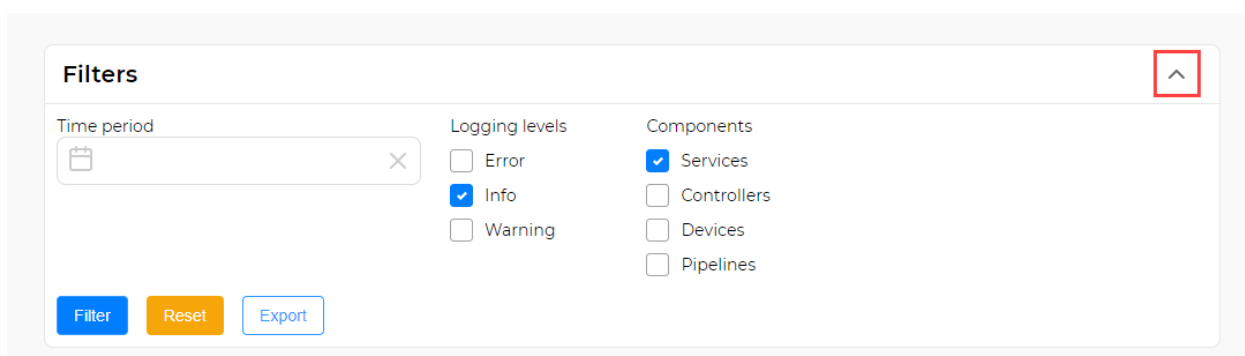


Figure 39. Hiding the filters

7. Access function

7.1. Import settings function

The **Import settings** function is designed to import settings from the local computer to the Access.

To import the settings, follow these steps:

1. Click on the arrow in the upper right corner ▼ to the right of the user's avatar to expand the drop-down menu (Figure 40).

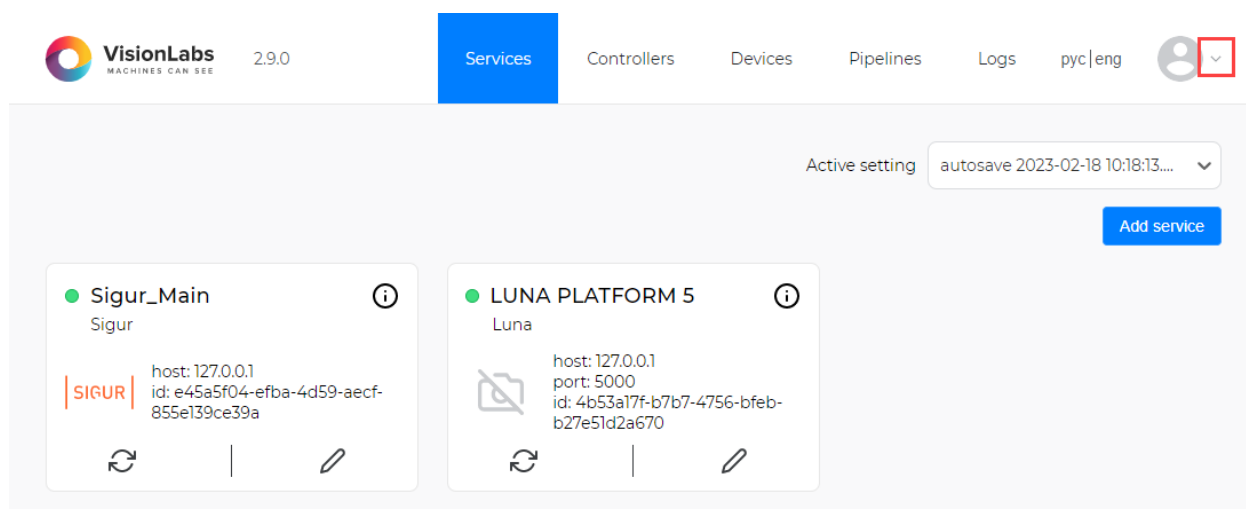


Figure 40. Switching to the **Import settings** function

2. Select the “Import Settings” function in the drop-down menu (Figure 41).

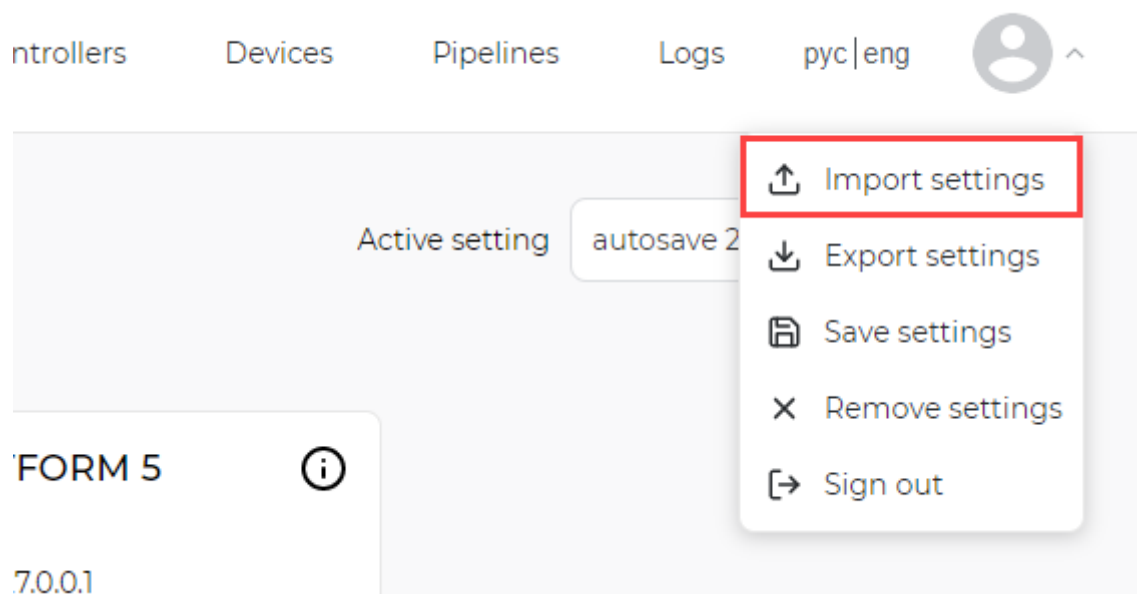


Figure 41. **Import settings** function

3. The file export form will open, click on **+** to select a json file (Figure 42).

The form is a light gray box containing a dashed box with a '+' icon and the text 'Setting file' next to it. Below this is a text input field labeled 'Setting name'. At the bottom of the form is a blue button labeled 'Upload'.

Figure 42. Selecting a settings file

4. In the “Setting name” field, the name of the imported file will be transferred; if necessary, specify the name of the loaded setting (Figure 43).

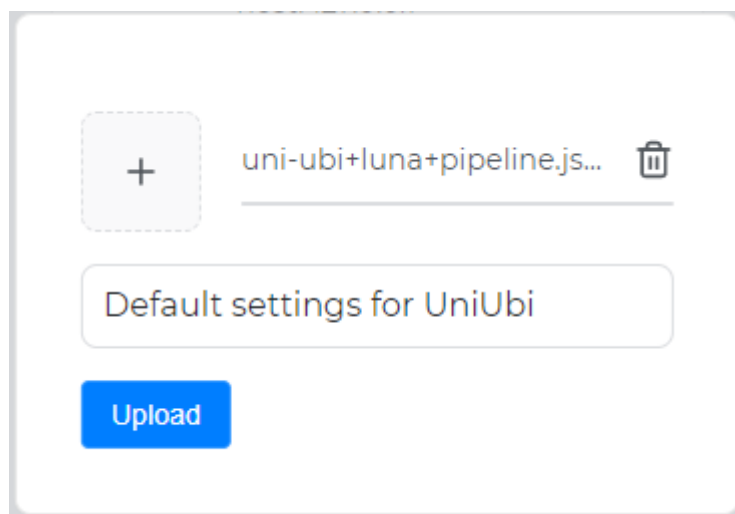


Figure 43. The **Setting name** field

5. Click on the **Add** button (Figure 44).

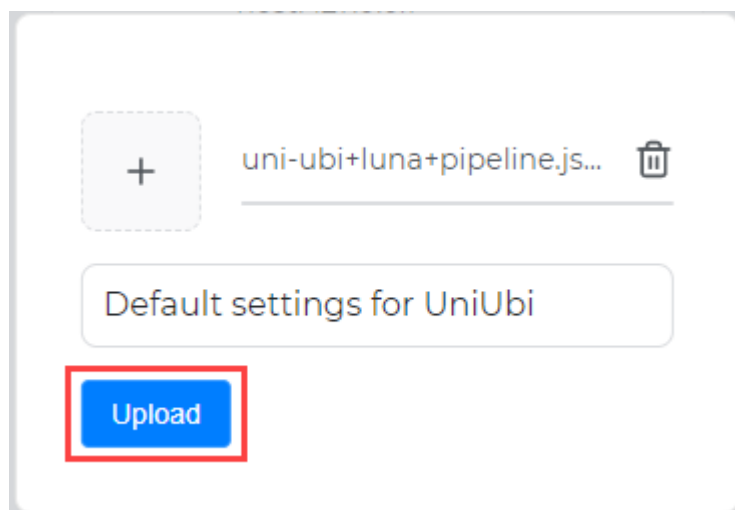


Figure 44. Adding a settings file

After successfully importing the setup, the message **Setup loaded** will be displayed in the upper-left corner of the screen (Figure 45).

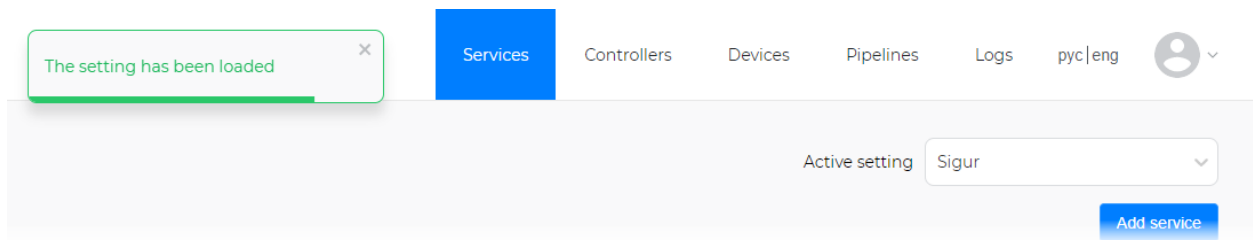


Figure 45. Setting upload confirmation

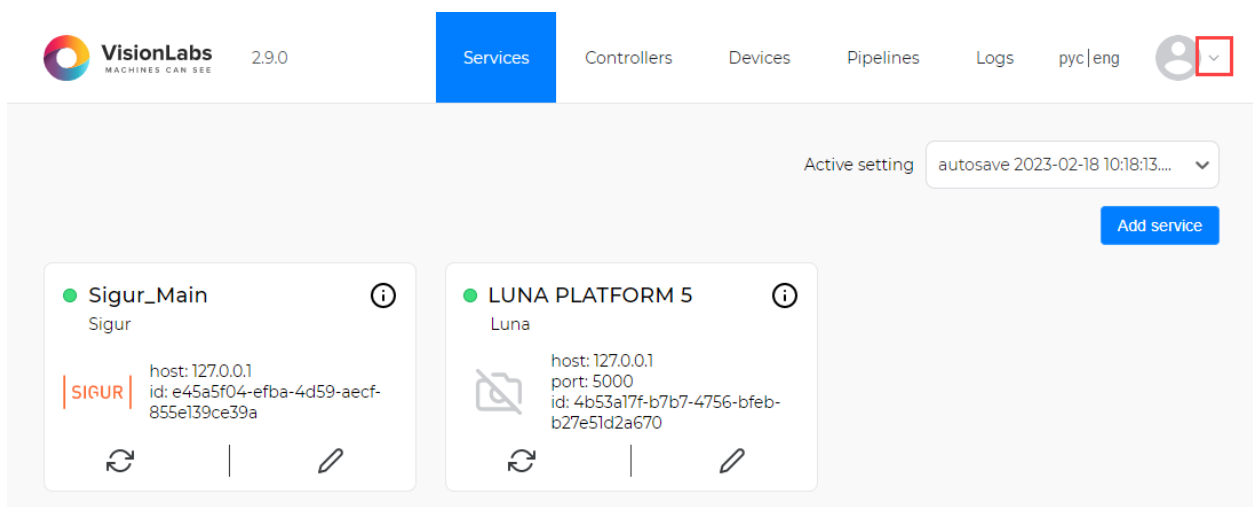
If the setting is created and imported correctly, the components contained in this setting will be displayed in the corresponding sections “Services”, “Controllers”, “Devices” and “Pipelines”.

7.2. Export settings function

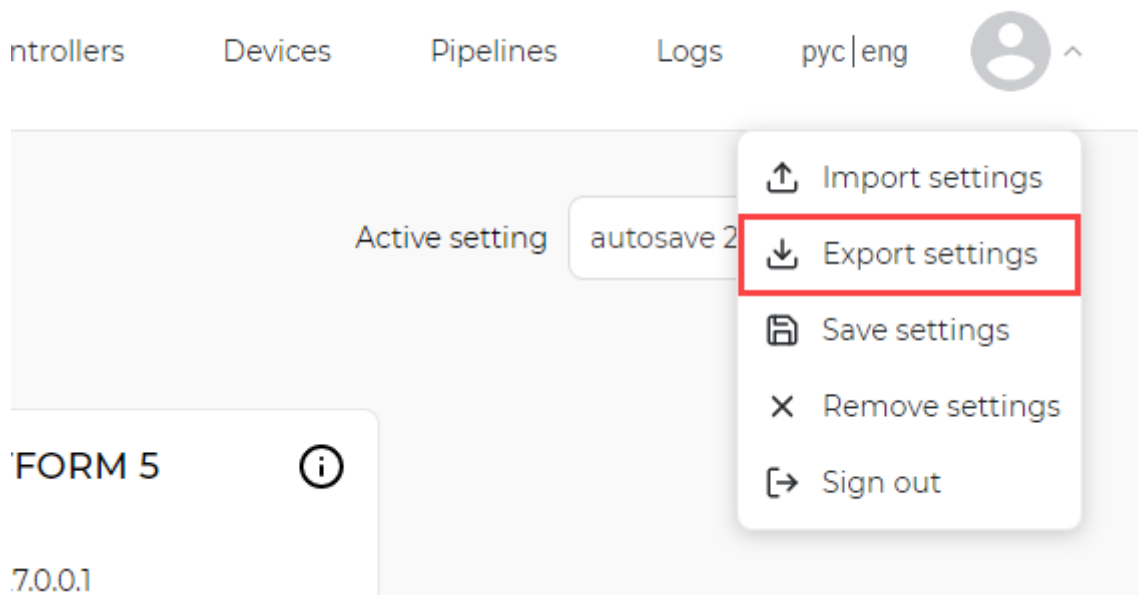
The **Export settings** function is designed to export settings to a local computer from the Access.

To export the settings, follow these steps:

1. Click on the arrow in the upper right corner ▼ to the right of the user’s avatar to expand the drop-down menu (Figure 46).

Figure 46. Switching to the **Export settings** function

2. Select the **Export Settings** function in the drop-down menu (Figure 47).

Figure 47. **Export settings** function

A file with the saved settings will be downloaded to the local computer in json format with all the components that are contained in this setting.

7.3. Reset Settings function

The **Reset Settings** function deletes all components and related data in the **Services**, **Controllers**, **Devices** and **Pipelines** sections.

7.4. Full name variables

The fields for entering `successful_pass_message_template` allow you to display the full name of recognized faces:

If the full name has more than 30 characters, then the first name will be shortened automatically to Last name with initials.

- {fullname} - full name of the person from the "Information" field (user_data). Ivanov Petr Sergeevich;
- {lastname} - Last name of the person. This is the first word from the "Information" field. Ivanov;
- {firstname} - The name of the person. This is the second word from the "Information" field. Peter;
- {middlename} - Middle name of the person. This is the third word from the "Information" field. Sergeevich;
- {short_lastname} - The first letter of the last name with a dot at the end. I.;

- {short_firstname} - The first letter of the name with a dot at the end. P.;
- {short_middlename} - The first letter of the patronymic with a dot at the end. S. .

To display the message on the terminal screen, it is required to specify the desired variants of full name in the `successful_pass_message_template` and `unsuccessful_pass_message` fields (Table 8).

Table 8. Example of using the variables full name

Record in settings	Output on terminal
Welcome, {fullname}!	Welcome, Ivanov Petr Sergeevich!
Welcome, {firstname} {middlename}!	Welcome, Peter Sergeevich!

7.5. Other functions

7.5.1. Documentation

The **Documentation** button is designed to switch to the online Access documentation.

8. Services

Services in Access are required to select options for connecting to external systems.

All fields are required unless otherwise stated in the description.

8.1. Apacs

This service is designed to interact with the [APACS 3000](#) ACS.

8.1.1. Apacs functionality

Main functions:

- adding devices that the biometric system will work with;
- receiving regular updates from the ACS software database;
- sending requests to add/change data in the local person storage;
- receiving identification events;
- sending a request to the ACS software about identification events;
- integration with LP5;
- integration with CBS: Alfa;
- logging of events about an attempt by an unidentified employee to pass through the turnstile.

8.1.2. Configuring parameters for connecting to the APACS ACS

Service settings and possible values (Table 9):

Table 9. Setting up the APACS service

Parameter	Description	Possible values	Default value
name	User-defined service name	Any textual names. Only Latin characters are supported. It is not recommended to enter more than 30 characters.	-
bio_system_id	The identifier of the biometric system	-	-
host	IP address of the server with installed APACS software	IP address in the form X.X.X.X	-

Parameter	Description	Possible values	Default value
port	Port on which APACS is deployed	-	7010
max_workers	Number of parallel threads for face replication	>0/10	
enable_ssl	Data encryption method for network transmission. It depends on the type of network you are using.	On - https Off - http	Off
login	Login of ACS software user. Input of Latin characters, numbers and symbols is supported.	User created in APACS ACS software	-
password	The user's password created in the APACS ACS software.	The input of Latin letters, numbers and symbols is supported.	-
feature_profile	The profile key belonging to the master key of the system. The key data is located in the APACS ACS software: Help → About the program	-	-
rabbitmq_login	Username from RabbitMQ from Apacs	The input of Latin letters and numbers is supported	-
rabbitmq_password	The password of the user from RabbitMQ from Apacs	The input of Latin letters and numbers is supported	-
card_format_source	The type of map format for uploading organization codes and their offsets. For more information, see below	-	-
enable_controller_creation	Enabling replication of ApacsController controllers	On - replication is enabled Off - replication is disabled	On

Parameter	Description	Possible values	Default value
card_priority_number	A priority marker for maps. All cards with the same priority number will have higher priority	Numeric values greater than or equal to 0	-
kyc_field_number	The number of the additional field in employee cards where internal employee IDs (KYC) are specified	1-20	-

The master key can be found in any Apacs client application in the Help → About section.

The type of card format (card_format_source) can be found in any Apacs client application in the Console tab → System Root section → Hardware Server section → select a network driver → select a controller → Group: Card Format section → select any card format → General tab → field “The type of the object”.

8.2. Bastion

The service is designed to interact with [Bastion ACS 2 and 3](#).

8.2.1. Bastion functionality

Main features:

- adding devices that the biometric system will work with;
- receiving regular updates from the ACS software database;
- sending requests to add/change data in the local person storage;
- receiving identification events;
- sending requests to the ACS software about identification events;
- integration with CBS MTS;
- logging events about an attempt by an unidentified employee to pass through the turnstile.

8.2.2. Bastion settings

The following settings are used when creating a new service (Table 10):

Table 10. Bastion Service Configuration

Parameter	Description	Values	Default value
name	Service name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
bio_system_id	Drop-down list for selecting the biometric system identifier (LP5 or CBS) in Access.	-	-
host	IP address of the server with installed Bastion software	IP address in the form of X.X.X.X or site.domain	-
port	ONVIF port of Bastion service	-	10112
enable_ssl	SSL encryption support for messages. It must be activated if necessary to maintain confidentiality. When activated, the load on the device and the message transmission time increases	On - https Off - http	Off
username	Bastion ONVIF user login. Input of Latin characters, numbers and symbols is supported.	The user created in Bastion	-
password	Bastion ONVIF user password. Input of Latin characters, numbers and symbols is supported.	User password	-
vl_access_host	IP address of the server where Access is installed	IP address in the form of X.X.X.X or site.domain	-
vl_access_port	Port of the server where Access is deployed	-	9091

Parameter	Description	Values	Default value
bastion_version	Dropdown list for selecting the Bastion ACS version	2 or 3	2

8.3. Bolid

The service is designed to interact with the [Bolid](#) ACS (../bolid.md#bolid-scud).

8.3.1. Bolid settings

The following settings are used when creating a new service (Table 11):

Table 11. Setting up the Bolid service

Parameter	Description	Values	Default value
name	Service name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
bio_system_id	Drop-down list for selecting the ID of the Luna service in Access.	-	-
host	IP address of the server with installed Bolid software	IP address in the form X.X.X.X	-
port	Bolid service port	-	8090
max_workers	Number of data replication handlers from the Luna list to Bolid. If there is a large amount of data, it is recommended to set from 2 to 5.	>0	10

Parameter	Description	Values	Default value
enable_ssl	SSL encryption support for messages. It must be activated if necessary to maintain confidentiality. When activated, the load on the device and the message transmission time increases	On - https	Off
		Off - http	
login	Username of the Bolid user. It is set in the Bolide software: DB → Passwords → The password type is "Remote control"	The user created in the bolid. The input of Latin letters, numbers and symbols is supported.	-
password	Bolid user password. Input of Latin characters, numbers and symbols is supported.	User password	-
token_ttl_sec	Time to refresh the access token (in seconds). Find the value of the TokenLifeTime field in the file ProgramData\ BolidIntegrServ\ settings.ini	It is not recommended to change this Parameter	300

8.4. CbsAkbars

Used to obtain a descriptor identifier in Akbars KBS from a photograph.

8.4.1. Configuring parameters for connecting to CbsAkbars

Service settings and possible values (Table 12):

Table 12. Configuring the CbsAkbars service

Parameter	Description	Possible values	Default value
name	Service name specified by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
host	IP address or LP5 domain name in CBS	IP address in the form X.X.X.X. or site.domain.	-
port	Port for connecting to LP5 in CBS	-	5000
enable_ssl	SSL encryption support for messages. Must be activated if privacy is required. When activated, the load on the device and the message transmission time increase	On – active	Off
		Off – inactive	
token	VisionLabs token for access to CBS.	-	-
match_class	Optional field for selecting one or more matching classes. The list of classes is specified by enumerating values separated by commas.	-	fz115_class
		fz115_class	
		import_high_class	
		import_low_class	
		selfreg_class	
		selfreg_wopass_class	

8.5. CbsAlpha

Used to obtain a descriptor ID in Alfa CBS based on KYC. Checks if a person exists in the CbsAlpha list by KYC with the ID specified in the `cbs_list_id` parameter, and creates a person if KYC is not found.

Person search by KYC is performed based on the value specified in the additional field 20 of the APACS ACS employee card. The number of this field is determined by the `kyc_field_number` parameter in the Apacs service.

KYC data is filled in the `external_id` field (External ID) in LUNA CLEMENTINE 2.0.

Only LUNA PLATFORM 5.10 and newer are supported.

8.5.1. Setting up parameters for connecting to CbsAlpha

Service settings and possible values (Table 13):

Table 13. Setting up the CbsAlpha service

Parameter	Description	Possible values	Default value
name	The name of the service specified by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
host	FRS IP address in KB	IP address in the form of X.X.X.X or site.domain.	-
port	Port for connecting to the FRS in CBS	-	5000
enable_ssl	SSL encryption support for messages. It must be activated if necessary to maintain confidentiality. When activated, the load on the device and the message transmission time increases	On – active	Off
		Off – inactive	
username	Username of the FRS user	-	-
password	Password of the FRS user	-	-
account_id	UUID of the FRS user	-	-

Parameter	Description	Possible values	Default value
handler_id	UUID of the handler for handling pass events, created in the FRS.	UUID of the handler	-
default_list_id	UUID of the ID of the FRS list that employees will be synchronized with	The ID of the list created in the FRS.	-
face_detection_threshold	The minimum threshold for face recognition is	0...1	0.5
event_receiving_mode	The mode for receiving events from LP5 (from version 5.53.0). Optional field	None - do not listen to events websocket protocol using a persistent connection webhook - HTTP callbacks. Client - Luna Platform, server - Luna Access	websocket
vl_access_host	IP address of the server on which Access is installed	IP address in the form X.X.X.X	-
vl_access_port	The port of the server where Access is deployed	-	9091
vl_access_basic_username	Login for interacting with Access	-	-
vl_access_basic_password	Password for interacting with Access	-	-
max_greatest_side_size	During replication, reduce the larger side of the photo to the specified size while maintaining proportions (Empty value - do not reduce photos)	0...1920	-
cbs_list_id	ID of the EBS list of persons	-	-

8.6. CbsAlphaListSynchronisation

The service is designed to synchronize two lists in CbsAlpha. The service tracks changes in the cbs list and matches them with entries from the luna list. If duplicate faces are found, the duplicate is removed from the luna list.

8.6.1. Configuring CbsAlphaListSynchronisation Settings

Service settings and possible values (Table 14):

Table 14. CbsAlpha service configuration

Parameter	Description	Possible values	Default value
name	Service name specified by the user	Any text names. Supports input of Latin and Cyrillic characters. It is not recommended to enter more than 30 characters.	-
pac_id	Unique service identifier in Access	Dropdown list for service selection	-
synchronisation_interval_hours	Frequency of launching synchronization between luna list and cbs list in hours	>0	-

8.7. CbsMts

Used to retrieve the descriptor identifier in the MTS CBS from a photo.

8.7.1. Configuring CbsMts settings

Service settings and possible values (Table 15):

Table 15. Setting up the CbsMts service

Parameter	Description	Possible values	Default value
name	User-defined service name	Any textual names. Only Latin characters are supported. It is not recommended to enter more than 30 characters.	-
host	IP address or domain name in KBS	IP address in the format X.X.X.X. or site.domain.	-
port	Port for connecting to CBS. The field can be empty	-	55580
enable_ssl	SSL encryption support for messages. It must be activated if necessary to maintain confidentiality. When activated, the load on the device and the message transmission time increases	On – active	Off
		Off – inactive	
urn	Path to the directory of persons in CBS	-	/cbs/persons
token	VisionLabs token for access to MTS CBS	-	-
timeout	Timeout time in seconds when connecting to the service fails. It is necessary to take time if there is a large delay between servers.	The time is selected taking into account the network delay to maintain performance.	10
cert_name	The name of the certificate to connect to the CBS. Certificate storage directory 'tls/	-	-

8.8. CbsVtb

Used to obtain the descriptor ID in the VTB CBS by photo.

8.8.1. Configuring parameters for connecting to CbsVtb

Service settings and possible values (Table 16):

Table 16. Configuring the CbsVtb service

Parameter	Description	Possible values	Default value
name	Service name specified by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
auth_host	IP address or domain name of the server for obtaining the VTB CBS authorization token	IP address in the form X.X.X.X. or site.domain.	-
auth_port	Server port for receiving the VTB CBS authorization token	-	-
auth_enable_ssl	SSL encryption support for messages. Must be activated if privacy is required. When activated, the load on the device and the message transfer time increase	On – active	Off
		Off – inactive	
auth_client_id	Client ID for receiving the VTB CBS token	-	-
auth_client_secret	Client key for receiving the token	-	-
host	IP address or domain name of the CBS	IP address in the form X.X.X.X. or site.domain.	-
port	Port for connecting to the CBS	-	5000

Parameter	Description	Possible values	Default value
enable_ssl	SSL encryption support for messages. Must be activated if privacy is required. When activated, the load on the device and the message transfer time increase	On – active	Off
		Off – inactive	
user_session_id	User session ID for sending photos for matching	-	-
ibm_client_id	ibm_client ID for sending photos for matching	-	-
system	VTB CBS system parameter. Requested from a VisionLabs representative	-	-
channel	VTB CBS system parameter. Requested from a VisionLabs representative	-	-
process_code	VTB CBS system parameter. Requested from a VisionLabs representative	-	-
signer_service_id	Drop-down list for selecting the CryptoPro service	Added CryptoPro service	-

8.9. CryptoPro

Used for signing or decrypting identification requests to a biometric system.

Main functions:

- certificate selection for signing;
- selection of the type of signature to be created;
- creation of a combined signature (contains the content to be signed and the signature);
- creation of a detached signature (signature only);
- signature verification.

8.9.1. Configuring parameters for connecting to CryptoPro

Service settings and possible values (Table 17):

Table 17. Configuring the CryptoPro service

Parameter	Description	Possible values	Default value
name	User-defined service name	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
host	IP address or domain name of the service	IP address in the form X.X.X.X. or site.domain.	-
port	Port for connecting to the service	-	-
api_key	Service access token	-	-
cert_serial_number	Serial number of the target certificate (you can get a list of installed certificates via API: http://host:port/docs)	-	-
cert_pin	PIN code of the target certificate	-	-
signature_type	Type of signature to create	PKCS7 CADES_BES	-

8.10. EyelsProxy

A proxy service that facilitates data transfer between the client and the controller: receives a request from the client, forwards it to the controller, obtains the response, and returns it back to the client.

8.10.1. Configuring parameters for connecting to EyelsProxy

Service settings and possible values (Table 18):

Table 18. Setting up the EyelsProxy service

Parameter	Description	Possible values	Default value
name	The service name specified by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
host	IP address or domain name of the Eyels controller	IP address in the format X.X.X.X or site.domain.	-
port	Port for connecting to the Eyels controller	-	-

8.11. Gate

The service is designed to interact with Gate PACS.

8.11.1. Configuring Gate settings

Service settings and possible values (Table 19):

Table 19. Gate Service Configuration

Parameter	Description	Possible values	Default value
name	User-defined service name	Any textual names. Only Latin characters are supported. It is not recommended to enter more than 30 characters.	-
luna_id	Drop-down list for selecting the Luna service identifier	-	-

8.12. Luna

The service is designed to redirect data from/to LP to external systems and devices.

Supported versions: 5.10 and higher.

8.12.1. Luna settings

The following settings are used when creating a new service (Table 20):

Table 20. Setting up the Luna service

Parameter	Description	Values	Default value
name	Service name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
host	IP address of the server where Luna is installed	IP address in the form of X.X.X.X or site.domain	-
port	Port of the server where Luna is deployed	-	5000
enable_ssl	Method of encrypting data during transmission over the network. Depends on the network type in the solution.	On - https	Off
		Off - http	
username	Username of the LP5 user. Filled in if account_id is not specified	-	-
password	The LP5 user's password. Filled in if account_id is not specified	-	-
account_id	UUID of the LP5 user. If specified, the username and password fields are not required	-	-
handler_id	UUID of the handler for working with passage events, created in Luna	-	-

Parameter	Description	Values	Default value
default_list_id	UUID of identifier the LP5 list with which the employees will be synchronized	The identifier of the list created in LP5.	-
face_detection_threshold	The minimum threshold for face recognition is	0...1	0.5
event_receiving_mode	The mode for receiving events from LP5 (from version 5.53.0). Optional field	None - do not listen to events websocket protocol using a persistent connection webhook - HTTP callbacks. Client - Luna Platform, server - Luna Access	websocket
vl_access_host	IP address of the server on which Access is installed	IP address in the form X.X.X.X	-
vl_access_port	The port of the server where Access is deployed	-	9091
vl_access_basic_username	Login for interacting with Access	-	-
vl_access_basic_password	Password for interacting with Access	-	-
max_greatest_side_size	During replication, reduce the larger side of the photo to the specified size while maintaining proportions (Empty value - do not reduce photos)	0...1920	-

8.13. LunaAceConverter

Service for sending data received from LUNA ACE devices to LP5. The received request from the device is redirected to the RRL, then a response is generated for the device based on the received response from

the LP5.

Supported version LUNA ACE 1.2.23

8.13.1. LuaAceConverter settings

The following settings are used when creating a new service (Table 21):

Table 21. Setting up the LUNA ACE service

Parameter	Description	Values	Default value
name	Service name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
luna_id	Drop-down list with Luna service IDs in Access	-	-

8.13.2. Setting up LUNA ACE

1. Connect to the device via SSH.
2. Open the file: `vi /opt/luna_ace/ace_device.conf`.
3. Specify the URL of the LuaAceConverter service in the `luna_platform_address` parameter.

To get the URL of the service, you need to go to the created LuaAceConverter service in Access and copy the full path from the browser search bar:

```
http://<ip_address>:9092/service/<UUID>
```

4. Change to the directory: `cd /opt/luna_ace/services/ace_device`
5. Restart the device: `restart`

8.14. LunaCars

Software and hardware integration required for communication between LUNA CARS and barriers (boom barriers, sliding gates, bollards, and others) for vehicle access control.

Supported LUNA CARS modules:

- LUNA CARS API: v.4.0.15;
- LUNA CARS Stream: v.3.0.20;
- LUNA CARS Analytics: v.4.0.8;

Access links to LUNA CARS Analytics backend.

Events in the queue are of type `CarDetectionEvent`.

8.14.1. LunaCars settings

The following settings are used when creating a new service (Table 22):

Table 22. Setting up the LunaCars service

Parameter	Description	Values	Default value
name	Service name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
host	IP address of the server where LUNA CARS is installed	IP address in the form of X.X.X.X or site.domain	-
port	Port of the server where LUNA CARS is deployed	-	8080
enable_ssl	Method of encrypting data during transmission over the network. Depends on the network type in the solution.	On - https	Off
		Off - http	
api_port	Port of the server where LUNA CARS API is deployed	-	8100
login	LUNA CARS Analytics administrator login. Input of Latin characters, numbers and symbols is supported.	-	admin@test.ru

Parameter	Description	Values	Default value
password	LUNA CARS Analytics administrator password. Input of Latin characters, numbers and symbols is supported.	-	admin
event_expiry_time	After how many seconds events can be skipped as obsolete. It is necessary to reduce the time to ~15 seconds if the vehicle flow is constant	>10	60
min_license_plate_accuracy	Minimum accuracy of vehicle registration plate recognition	The value is formed at the design stage and corrected at the testing stage (0,00...1,00)	0,6
event_memory_time	Time during which the service does not create a repeat event for the same vehicle (in seconds). It is necessary to increase the value if the vehicle stands in the recognition zone for a long time in the queue for entry, etc.	60...180	90
timeout	Timeout for an unsuccessful attempt to connect to the service. It is necessary to increase the time if there is a large delay between servers	The time is selected based on the delay in the network to maintain performance	-

8.15. LunaStreams

A service for working with LunaStreams.

LUNA Streams is a service of VisionLabs FaceStream.

The service is designed to:

- receive a list of stream names from LUNA Stream for subsequent transmission to the ACS;
- generating a detection event based on a frame from LUNA Stream for subsequent sending for matching to Luna, CbsMts or CbsAlpha.

Supported version is:

- FaceStream 5.1.6 or newer;
- LunaStreams 0.2.1 or newer.

8.15.1. Configuring LunaStreams settings

Service settings and possible values (Table 23):

Table 23. Setting up the LunaStreams service

Parameter	Description	Possible values	Default value
name	User-defined service name	Any textual names. Only Latin characters are supported. It is not recommended to enter more than 30 characters.	-
host	IP address of the server with installed LunaStreams	IP address in the form X.X.X.X	-
port	The port of the server where LunaStreams is deployed	-	5160
enable_ssl	Method of encrypting data during transmission over the network. Depends on the network type in the solution.	On - https	Off
		Off - http	
handle_event_interval	The delay interval between receiving a detection from a single source.	1...10	3

8.16. Parsec

This service is designed to interact with the [Parsec](#) ACS to ensure the passage of recognized persons through a turnstile/door with a magnetic lock.

8.16.1. Parsec functionality

Main features:

- receiving information about access points;
- receiving regular updates from the ACS software database;
- sending requests to add/edit/delete data in the local person storage;
- receiving identification events;
- sending a request to the ACS software about identification events;
- integration with MTS KBS and LP 5;
- logging events about an attempt by an unidentified employee to pass through the turnstile.

8.16.2. Parsec settings

The following settings are used when creating a new service (Table 24):

Table 24. Setting up the Parsec service

Parameter	Description	Values	Default value
name	Service name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
bio_system_id	Drop-down list for selecting the biometric system identifier (LP5 or CBS) in Access.	-	-
host	IP address of the server where Parsec is installed	IP address in the form of X.X.X.X or site.domain	-
port	Port of the server where Parsec is deployed	-	-

Parameter	Description	Values	Default value
enable_ssl	Method of encrypting data during transmission over the network. Depends on the network type in the solution.	On - https Off - http	Off
username	Parsec user login. Input of Latin characters, numbers and symbols is supported.	-	-
integration_key	The Parsec integration key. It is used as a password to connect to the service.	-	-
vl_access_host	IP address of the server where Access is installed	IP address in the form of X.X.X.X or site.domain	-
vl_access_port	Port of the server where Access is deployed	-	9091

8.17. PercoWeb

The service is designed for interaction with the [PERCo-Web] ACS (../perco.md#percoweb-scud).

8.17.1. PercoWeb functionality

Main features:

- adding devices that LP5 will work with;
- receiving regular updates from the ACS software database;
- sending requests to add/change data to LP5;
- receiving identification events;
- sending requests to the ACS software about identification events;
- logging events about an attempt by an unidentified employee to pass through the turnstile.

8.17.2. PercoWeb settings

The following settings are used when creating a new service (Table 25):

Table 25. Setting up the PercoWeb service

Parameter	Description	Values	Default value
name	Service name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
bio_system_id	Biometric system identifier	-	-
host	IP address of the server where PERCo-Web is installed	IP address in the form of X.X.X.X or site.domain.	-
port	Port of the server where PERCo-Web is deployed	-	-
enable_ssl	Method of encrypting data during transmission over the network. Depends on the network type in the solution.	On - https Off - http	Off
login	PERCo-Web user login. Input of Latin characters, numbers and symbols is supported.	The user created in PERCo-Web	-
password	PERCo-Web user password. Input of Latin characters, numbers and symbols is supported.	User password	-
token_ttl_min	The validity period of the security token. The value must match the PERCo-Web software, location of the PERCo-Web Manager → Settings → Advanced Settings → The lifetime of the session.	Minutes 0...10000	1440 (1 day)

Parameter	Description	Values	Default value
-	-	max_workers	The maximum number of threads that can be used for face replication

8.18. PersonStorageActualization

The service periodically updates the data in the person storage (PersonStorage). It removes a person from the person storage if they were not found in the biometric system.

PersonStorage stores information about employees in the ACS and their descriptor_id from the biometric system.

8.18.1. PersonStorageActualization Settings

Service settings and possible values (Table 26):

Table 26. PersonStorageActualization service configuration

Parameter	Description	Possible values	Default value
name	Service name specified by the user	Any text names. Supports input in Latin and Cyrillic. It is not recommended to enter more than 30 characters.	-
pacs_id	Unique service identifier in Access	Dropdown list for service selection	-
actualization_interval_hours	Person actualization interval in hours	>0	1

8.19. Rusguard

The service is designed to interact with ACS [RusGuard](#).

8.19.1. Rusguard functionality

Main functions:

- getting regular updates from the database BY ACS;
- sending requests to add/change data in the local person storage;
- receiving identification events;
- sending a request to the ACS software about identification events;
- logging events about an unidentified employee's attempt to pass through the turnstile;
- integration with CBS is possible.

8.19.2. Configuring settings for connecting to Rusguard

Service settings and possible values (Table 27):

Table 27. Configuring the Rusguard service

Parameter	Description	Possible values	Default value
name	The name of the service specified by the user	Any text names. Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 30 characters.	-
bio_system_id	Biometric system identifier	-	-
host	IP address or domain name of the server with Rusguard installed	IP address in the form of X.X.X.X. or site.domain.	-
port	The port of the server where Rusguard is deployed	-	8089
max_workers	The maximum number of streams that can be used for face replication.	>0	10

Parameter	Description	Possible values	Default value
enable_ssl	Method of data encryption during network transmission. Depends on the type of network used.	On - https Off - http	Off
target_photo_number	The number of the photo in the ACS that is used for replication.	0...100	1
target_card_type_id	ID of the replicated card type. If the field is empty, any employee card is replicated. The available IDs of the card types are displayed in the Info block.	-	-
replicate_session_interval_sec	The frequency of synchronization of the ACS database and Access storage. You must specify the minimum allowed synchronization time, since Access does not receive notifications from external systems about adding/removing an employee.	Set in seconds 0...100/5	

8.20. Salto

The service is designed to interact with the [SALTO](#) ACS.

8.20.1. Configuring settings for connecting to Salto

When creating a new service, the following settings are used (Table 28):

Table 28. Configuring the Salto service

Parameter	Description	Possible values	Default value
name	The name of the service specified by the user	Any text names. Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 30 characters.	-
bio_system_id	Biometric system identifier	-	-
host	IP address or domain name of Salto.	-	-
port	The port of the server where Salto is deployed.	-	8100
enable_ssl	Method of data encryption during network transmission. Depends on the type of network in the solution.	On - https	Off
		Off - http	
login	Login of the Salto user. Input of Latin letters, numbers, and symbols is supported.	User created in Salto	-
password	The password of the Salto user. Input of Latin letters, numbers, and symbols is supported.	User's password	-
max_workers	The maximum number of streams that can be used for face replication.	>0	10

8.21. Sigur

The service is designed to interact with the [Sigur](#) ACS.

8.21.1. Sigur functions

Main functions:

- adding devices with which LP5 and LUNA CARS will work;
- receiving regular updates from the ACS software database;
- sending requests to add/change data in LP5;
- receiving identification events;
- integration with CBS: MTS, VTB, Ak Bars;
- sending a request to the ACS software about identification events.
- logging events about an attempt by an unidentified employee to pass through the turnstile.

8.21.2. Setting parameters for connecting to Sigur ACS

To add a service, you need to create it with the following settings (Table 29):

Table 29. Setting up the Sigur service

Parameter	Description	Possible values	Default value
name	Service name specified by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
bio_system_id	Drop-down list for selecting the biometric system identifier (LP5 or CBS) in Access.	-	-
host	IP address or domain name of Sigur.	IPv4 or site.domain	-
luna_cars_id	LunaCars service identifier in Access.	-	-
mark_for_ignore	When synchronizing with Sigur, if this combination is found in the body of an employee's request, the request is ignored. This option is necessary for separating different copies of Sigur in one system.	Any string value	-

Parameter	Description	Possible values	Default value
ignore_ replication_field_ name	The name of an additional field responsible for ignoring during replication. The field must be of a logical type and answer the question: Ignore the employee during replication? (Yes/No)	Any string value	-
additional_ person_field	Name of the additional field from the employee card where the descriptor ID is written	Any string value	-

8.22. SigurThroughDatabase

The service provides integration with the Sigur ACS through a direct connection to its database.

Access synchronizes employee data from the Sigur database, taking into account only those users who have an access card.

Upon successful identification of an employee in the Luna biometric system, Access sends his card number to the GateController or PusrController intermediate controllers physically connected to the Sigur ACS. These controllers transmit commands to open/close the passage.

Integration is carried out directly with the Sigur database, without using the API

Sigur does not initiate a connection with Access — data is transmitted only in one direction (from Access to Sigur via controllers)

Sigurthdatabase functionality

Main functions:

- receiving regular updates from the database on ACS;
- sending requests for adding/changing data to the SRL;
- receiving identification events;
- sending a request to the ACS software about identification events;
- logging of events about an attempt by an unidentified employee to pass through the turnstile.

8.22.1. LP5 Integration Options

Each LP5 integration (Table 30) uses the [Luna](#) service.

Table 30. LP5 Integration Options

Service	Device	Pipeline
SigurThrough Database + PusrController / GateController	Beward	MatchByPhoto + SendToDevice + SendToController
	BioSmart	MatchByPhoto + SendToDevice + SendToController
	Dahua	MatchByPhoto + SendToController
	Dahua Thermo	MatchByPhoto + SendToController
	Fortuna315	MatchByPhoto + SendToController
	HikvisionCamera	MatchByPhoto + SendToController
	HikvisionCamera Thermo	MatchByPhoto + SendToController
	HikvisionTerminal Thermo	MatchByPhoto + SendToDevice + SendToController
	LunaFast4A1	MatchByPhoto + Custom2FA
	Panda	MatchByPhoto + SendToController
	UniUbi	MatchByPhoto + SendToDevice + SendToController
	VKVision02	LunaStreams + MatchByPhoto + SendToDevice + SendToController
	R20Face	MatchByPhoto + SendToDevice + SendCardToR20Face

8.22.2. Configure settings for connecting to Sigurthdatabase

When creating a new service, the following settings are used (Table 31):

Table 31. Configuring the Sigurthdatabase service

Parameter	Description	Possible values	Default value
name	The name of the service specified by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
bio_system_id	Biometric system identifier	-	-
host	IP address or domain name of the Sigur database.	IP address in the form of X.X.X.X. or site.domain.	
port	Sigur database port.	-	3305
-	-	max_workers	The maximum number of threads that can be used for face replication
db_username	Username used to connect to the Sigur database	-	-
db_password	Sigur database user password	-	-

8.23. Strazh

The service is designed to interact with the [STRAZH](#) ACS (../strazh.md#strazh-scud).

- Supports integration with CBS MTS.

8.23.1. Strazh settings

To add a service, you must create it with the following settings (Table 32):

Table 32. Configuring the Strazh service

Parameter	Description	Values	Default value
name	Service name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
bio_system_id	Biometric system identifier	-	-
login	Strazh user login. Input of Latin characters, numbers and symbols is supported.	The user created in Strazh	-
password	Strazh user password. Input of Latin characters, numbers and symbols is supported.	User password	-
host	IP address of the server where Strazh is installed	IP address in the form of X.X.X.X or site.domain.	-
port	Port of the server where Strazh is deployed	-	443
enable_ssl	Method of encrypting data during transmission over the network. Depends on the network type in the solution.	On - https Off - http	Off
max_workers	The maximum number of threads that can be used for face replication.	1-10	10
additional_person_field	Name of the additional person field from the ACS, where the descriptor identifier will be written.	-	-

8.24. Ubs

The service receives and processes messages from Kafka in the Moscow regional segment of the EBS, with the final processing result being the creation of an identification event.

8.24.1. Ubs settings

Service settings and possible values (Table 33):

Table 33. Ubs service setup

Parameter	Description	Possible values	Default value
name	Service name specified by the user	Any text names. Supports input in Latin and Cyrillic. It is not recommended to enter more than 30 characters.	-
pacs_id	Unique service identifier in Access	Dropdown list for service selection	-
kafka_servers	List of IP addresses or URLs separated by commas, for connecting to Kafka EBS	-	
kafka_topic	Kafka topic to listen to	-	
project_id	Project identifier in EBS	-	
event_expiry_time	Event validity time in seconds	60	

9. APACS ACS

Software integrations of the APACS ACS with biometric systems are implemented to ensure the passage of recognized persons through a turnstile/door with a magnetic lock.

- The supported version of the APACS ACS is 8.3.1.0.

Connection of AAM LAN 8W and AAN controllers is supported.

The ability to run multiple instances of Luna Access integrated with a single APACS ACS is supported.

9.1. Supported integration options for APACS ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

Transfer of user data from the ACS to LP5 occurs using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

For the synchronization/replication settings, see the service settings.

Each integration with LP5 (Table 34) uses the [Luna](#) service.

If the terminal does not have data output facilities (e.g., a screen), the [SendToDevice](#) pipeline is not required.

Table 34. LP5 integration options

Service	Device	Pipeline
APACS + ApacsController	LunaFast4A1	SendToLuna + Apacs2FA / MatchByPhoto + SendToDevice + SendToController
APACS	GrgFaster	MatchByPhoto + SendToGrgFaster

Each integration with CBS (Table 35) uses the CBS service.

Table 35. CBS integration options

Service	Device	Pipeline
CbsAlpha + Apacs + ApacsController	Beward	MatchByPhotoInCbsAlpha + SendToController + SendToDevice
	Dahua	MatchByPhotoInCbsAlpha + SendToController
	HikvisionCamera	MatchByPhotoInCbsAlpha + SendToController
	LunaFast4A1	MatchByPhotoInCbsAlpha + SendToController + SendToDevice
	UniUbi	MatchByPhotoInCbsAlpha + SendToController + SendToDevice
	R20Face	MatchByPhotoInCbsAlpha + SendCardToR20Face + SendToDevice
	HikvisionCamera	MatchByPhotoInCbsAlpha + SendToController

9.2. Standard integration using Apacs

Integration 1f (Figure 48) and (Table 36).

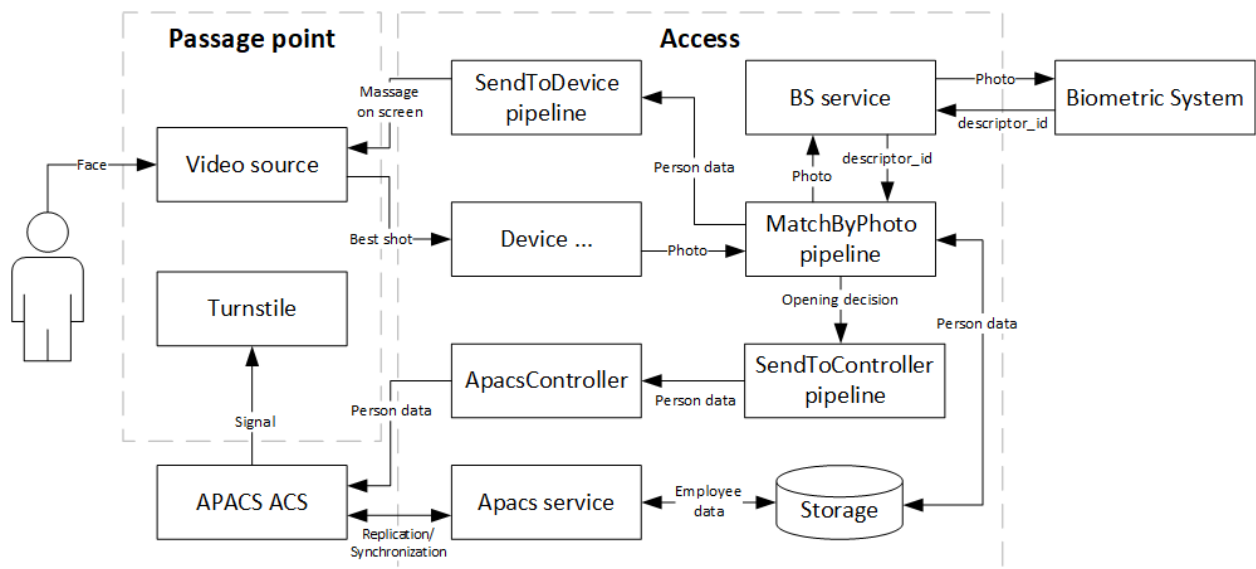


Figure 48. Component diagram for 1f integration

Table 36. Integration description

Component	Description
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Video source	A device for extracting a frame of a person's face. Can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream. A biometric terminal allows you to create feedback to show a person information about the passage.
Device ...	An Access component for receiving data from a video data source. Selected based on the device used.
MatchByPhoto pipeline	Access component for interacting with the BS. When working with a biometric terminal, it is necessary to additionally connect the SendToDevice
Biometric system	A system for comparing a reference photo of a person with the best frame obtained from a video data source. Can be either Luna or a service supported by CBS).
Apacs service	An Access component for replicating/synchronizing employees from the ACS and listening to ACS events.
SendToController pipeline	Access component for sending the card number and full name to the ApacsController after matching the person and confirming the card number in Access.
ApacsController	Access component for sending a card number to the ACS. When using the gate or pusher controller, the corresponding component must be used. When using a biometric terminal, it sends the employee's full name to it for display on the screen.
APACS ACS	Central software for working with Apacs. Stores employee data and makes a decision on granting access.
Turnstile	Barrier device for access control

Integration 2f (Figure 49) and (Table 37).

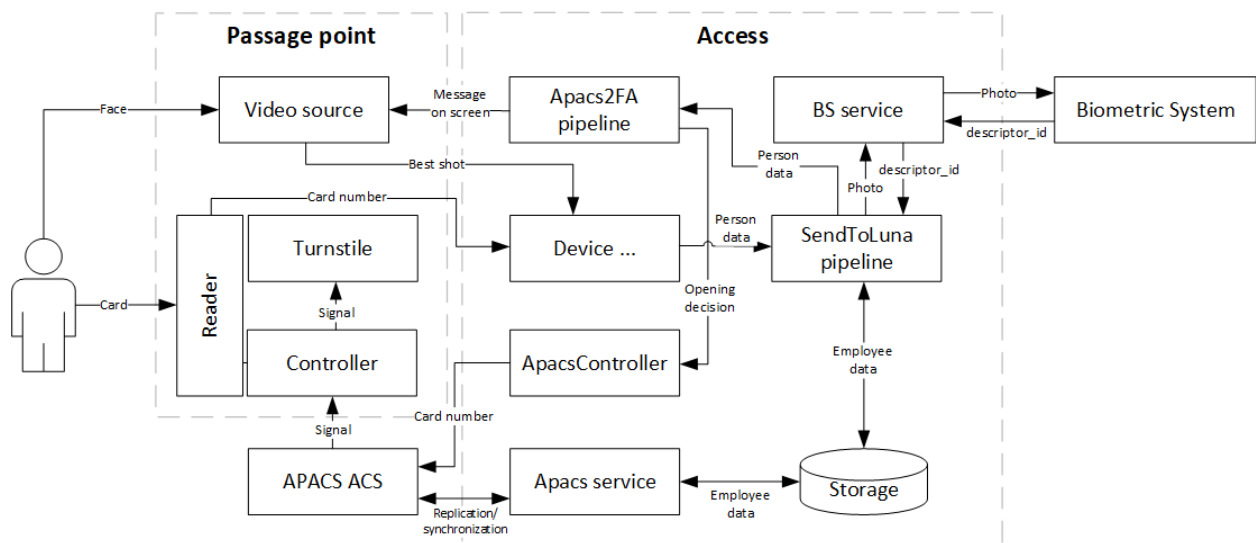


Figure 49. Component diagram for 2f integration with Apacs

Table 37. Integration description

Component	Description
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Reader	A device for receiving access card data.
Video source	A device for extracting a frame of a person's face. Can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream.
Device ...	Access component for receiving data from a video data source. Selected based on the device used.
SendToLuna pipeline	Access component for sending photos to LP5.
Luna service	Access component , which listens for matching events from LP5.
LP5	Biometric system for comparing a reference photo of a person and the best frame received from a video data source.
Apacs2FA pipeline	Access component , which receives a card number event and a person matching event. Compares the number received from the device with the number corresponding to the person and, if they match, passes the card number to ApacsController.

Component	Description
ApacsController	Access component for sending card numbers to the ACS. When using the gate or pusr controller, the corresponding component must be used.
Apacs Service	Access component for replicating/synchronizing employees from the ACS and listening to ACS events.
APACS ACS	Central software for working with Apacs. Stores employee data and makes decisions about granting access.
Turnstile	Barrier device for access control

9.2.1. Guest pass with two-factor authentication

If you need to pass a guest with active two-factor authentication and no possibility to get a guest photo, you need to perform the following steps:

1. Add a person to the ACS without a photo with full name and card number.
2. Enable the `use_cards_without_face` option in the [Apacs2FA](#) pipeline settings.

9.2.2. Creating a user in RabbitMQ

To create a user in RabbitMQ with read- and write-only rights to a specific queue, follow these steps:

Available on the server where Apacs is deployed.

1. Open the Windows command prompt.
2. Navigate to the directory of executable files for RabbitMQ:

```
cd "c:\Program Files\RabbitMQ Server\rabbitmq_server-*.*.*\sbin"
```

Substitute the value of the RabbitMQ Server version instead of the `*` character.

3. Add the user to RabbitMQ:

```
rabbitmqctl add_user <login> <password>
```

Substitute your own values instead of `<login>` and `<password>`. For more information, see the [official website](#).

4. Add rights to the user:

```
rabbitmqctl set_permissions -p / <login> "^apc.webapi.vl-access-2" "^apc.webapi.vl-access-2" "^apc.webapi.vl-access-2"
```

Substitute your own value instead of <login>. For more information, see the [official website](#).

These rights make it possible to create an exchange for the Apacs ACS in order to queue events about user changes. These rights also allow you to read events from this queue for further synchronization of users for Access.

9.3. Methods of interaction with Apacs

Beginning of endpoint for all requests (Table 38): /v1/webapi/v3.

Table 38. APACS methods

Task	Method	Description
Log in	POST /session/login/	Access authorization in ACS. Authorization occurs when adding a service and before logging out of the system
Log out of the system	POST /session/logout/	Sent when restarting or deleting the Apacs component
Get information about ACS	GET /webapi/ping/	Checking ACS availability once per minute.
Create a request	POST /query/	Create a request (getting employee data, card number, etc.) and get the request ID (query_id)
Get query result	POST /query/{query_id}/500/	Request to get result (500 - number of objects), with object ID (object_id).
Get data by ID	POST /object/id/{object_id}/	Get employee data
Send AAM card	POST /object/execCmd/{object_id} /cmdEmulateCardByNumber,	Send card to AAM/AAN controller
Send Apollo card	POST /object/execCmd/{object_id} /cmdSendCard/	Send card to Apollo controller

9.4. Apacs interaction process diagrams

9.4.1. Connecting the Apacs service

Sequence diagram (Figure 50).

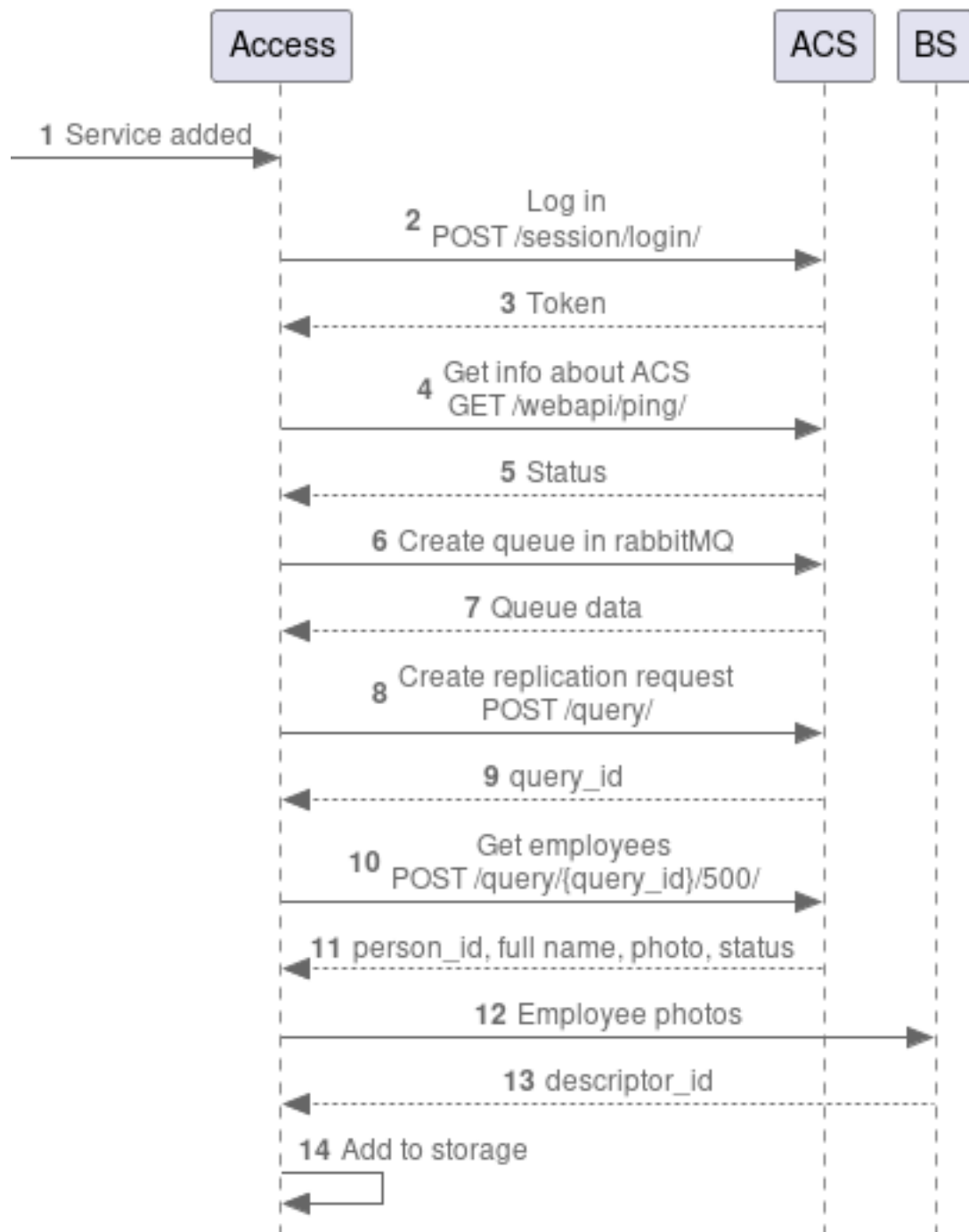


Figure 50. Process diagram for connecting the ACS

1. The user added the Apacs service to Access.
2. Access sends an authorization request to the ACS.
3. The ACS returns a token for authorization. The token has a lifetime, after which Access re-performs authorization.
4. Access sends a request to obtain information about the ACS.
5. The ACS returns information. Access uses only the ACS version to check compatibility and user information in the UI.
6. Access sends a request to create a queue in rabbitMQ to view employee events.
7. The ACS returns the queue ID.
8. Access creates a request to replicate employees from the ACS.
9. ACS returns query_id.
10. Access sends a request to obtain results for the replication request.
11. ACS returns person_id, full name, status, photo, date and time of the last change.
12. Access sends a request with employee photos to the BS to extract descriptor_id (face_id).
13. BS returns descriptor_id.
14. Access saves information on each employee in local storage.

9.4.2. Processing Apacs events with 1 factor

Sequence diagram (Figure 51).

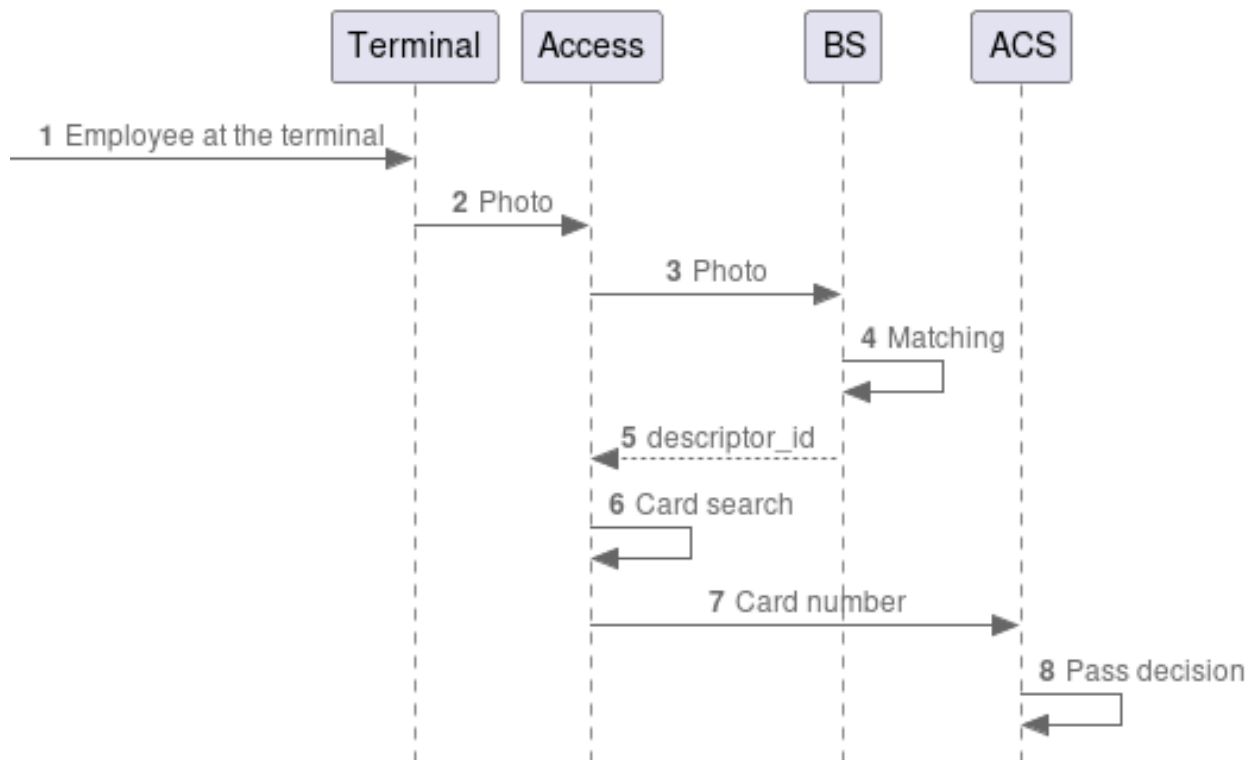


Figure 51. Process diagram with 1 factor

1. Employee at the biometric terminal at the checkpoint.
2. The terminal sends the best shot of the employee to Access.
3. Access sends a photo of the employee to the Biometric System.
4. The BS compares the photo from the terminal and the one saved in the database.
5. The BS returns the matching result to Access.
6. Access compares the face card number and the card number received from the employee.
7. Access sends the card number to the ACS.
8. The ACS makes a decision to allow the person through.

9.4.3. Processing Apacs events with 2 factors

Sequence diagram (Figure 52).

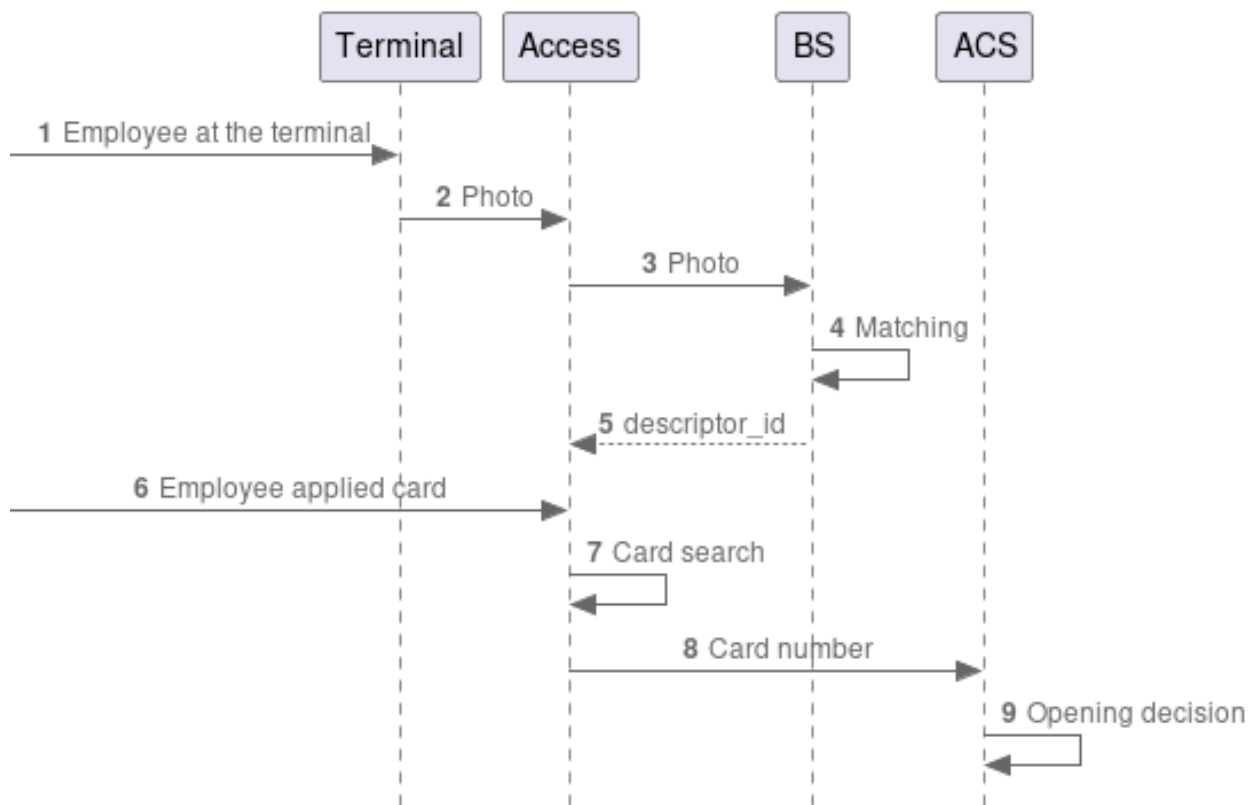


Figure 52. Process diagram with 2 factors

1. Employee at the biometric terminal at the checkpoint.
2. The terminal sends the best photo of the employee to Access.
3. Access sends the employee's photo to the Biometric System.
4. The BS compares the photo from the terminal and the one saved in the database.
5. The BS returns the matching result to Access.
6. Employee applies the card (the card use subprocess does not depend on photo processing, but, as a rule, the photo arrives first).
7. Access compares the face card number and the card number received from the employee.
8. Access sends the card number to the ACS.
9. The ACS makes a decision on whether to allow the person through.

9.5. Apacs FAQ

1. What is Facility_code?

Card numbers in Apacs have an offset that includes the organization number.

Access automatically cuts off the facility code when comparing card numbers (for example, an employee has a card 070.56458, in the ACS it is entered as 156458, where 1 is the offset).

2. The person has not replicated, what should I do?

If an employee replication error appears in the log during replication or the employee is not in the LP5 list (can be seen via LUNA CLEMENTINE), although he is in the ACS, then you need to go to the employee data > Accesses in the ACS, disable Activity and save the changes. Then enable Activity and save the changes.

10. Bastion ACS

The ACS synchronizes employees with the local person storage and listens to events, based on which it decides whether or not to open the turnstile. These events are generated in Access by the CreateBastionEvent pipeline.

- Supports Bastion ACS version: 2.1.11.2337, 2.1.13.2347, 2025.1, 0.23.3-17064.

For integration with **Bastion 2**, it is necessary to use pipelines that invoke text display on the terminal (it is recommended to use `SendToDevice`, or `LunaEventListener` when working with Luna).

In integration with **Bastion 3**, separate pipelines for text rendering are not required - this function is performed by `CreateBastionEvent`.

When connecting devices, it is necessary to specify the names of access points automatically generated by the service based on the access control system access points. They are specified in the Info service. They are generated in the format “access point name - identifier”. For example: “turnstile_exit - 907efa78-cb2f-4f46-b374-785c7f9901a5”.

The received access point names must be specified in:

- When using internal Access devices (HikvisionTerminal, Panda ...), specify in the “name” field
- When using LunaStream, specify in the “source” field

10.1. Supported integration options for Bastion ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

Transfer of user data from the ACS to LP5 occurs using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

For the synchronization/replication settings, see the service settings.

Each integration with LP5 (Table 39) uses the [Luna](#) service.

If the terminal does not have data output facilities (e.g., a screen), the [SendToDevice](#) pipeline is not required.

Table 39. LP5 integration options

Service	Device	Pipeline
Bastion	Beward	CreateBastionEvent + MatchByPhoto + SendToDevice
	BioSmart	CreateBastionEvent + LunaEventListener + SendToLuna
	Dahua	CreateBastionEvent + LunaEventListener + SendToLuna
	Dahua Thermo	CreateBastionEvent + LunaEventListener + SendThermalEventToLuna
	Fortuna315	CreateBastionEvent + LunaEventListener + SendThermalEventToLuna
	HikvisionCamera	CreateBastionEvent + LunaEventListener + SendToLuna
	HikvisionCamera Thermo	CreateBastionEvent + LunaEventListener + SendThermalEventToLuna
	HikvisionTerminal Thermo	CreateBastionEvent + LunaEventListener + SendThermalEventToLuna
	LunaFast4A1	CreateBastionEvent + LunaEventListener + SendToLuna
	Panda	CreateBastionEvent + LunaEventListener + SendThermalEventToLuna
	UniUbi	CreateBastionEvent + LunaEventListener + SendThermalEventToLuna/ SendToLuna
	VKVision02	CreateBastionEvent + LunaEventListener
	R20Face	CreateBastionEvent + LunaEventListener + SendToLuna

Each integration with CBS (Table 40) uses the CBS service.

Table 40. CBS integration options

Service	Device	Pipeline
CbsMts + Bastion	Beward	MatchByPhoto + SendToDevice + CreateBastionEvent
	Dahua	MatchByPhoto + CreateBastionEvent
	HikvisionCamera	MatchByPhoto + CreateBastionEvent
	LunaFast4A1	MatchByPhoto + SendToDevice + CreateBastionEvent
	UniUbi	MatchByPhoto + SendToDevice + CreateBastionEvent

10.2. Standard integration using Bastion

Bastion ACS software integrations with biometric systems are implemented to ensure the passage of recognized persons through a turnstile/door with a magnetic lock.

Bastion integration scheme for the passage of recognized faces through a turnstile/door with a magnetic lock. Standard Access components (Figure 53) and (Table 41) are used when integrating with Bastion.

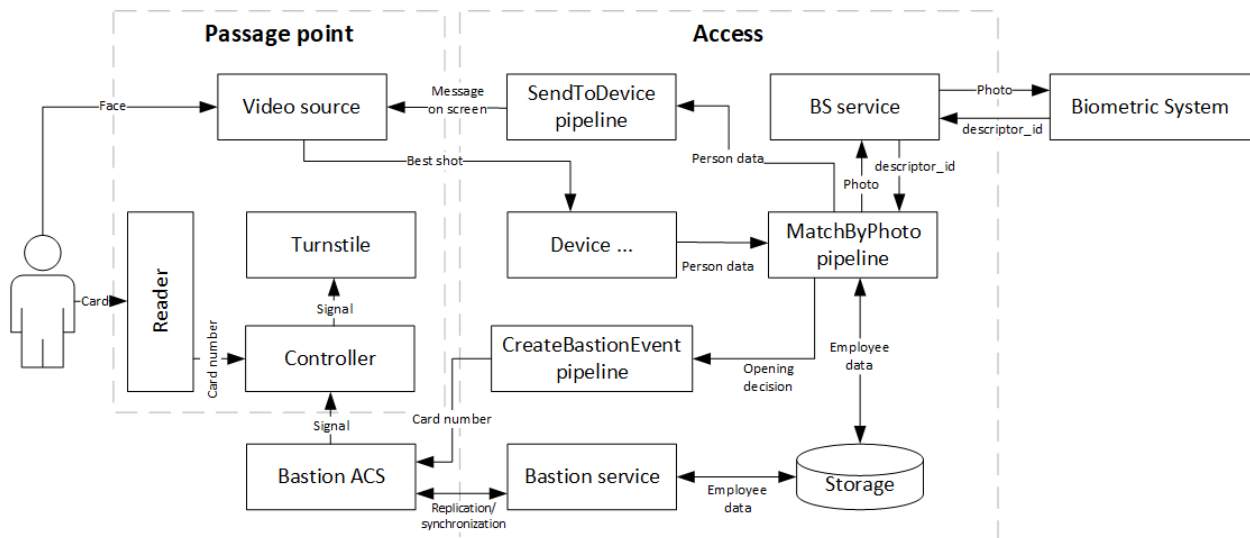


Figure 53. Component diagram for 1f integration

Table 41. Integration Description

Component	Description
1f	
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Video data source	A device for extracting a frame of a person's face. Can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream.
Device ...	An Access component for receiving data from a video data source. Selected based on the device used.
Controller	Passage point control board.

Component	Description
Turnstile	A barrier device for access control
Bastion ACS	Central software for working with Bastion. Stores employee data and makes decisions on access provision.
Bastion Service	Access component for processing information from ACS.
Add-on for 2f	
Reader	Device for receiving access card data.
Working with LP5 and CBS	
MatchByPhoto Pipeline	Access Component for interacting with BS
CreateBastionEvent Pipeline	Access Component for listening to event queues in Luna and generating events in Access
SendToDevice Pipeline	Access Component for sending a signal to open a relay to a device and displaying text on the screen. Required only when integrating Bastion 2 ACS

10.3. Setting up the Bastion 3 ACS software

1. Open the Bastion-3 ACS software — Control Panel.
2. Go to the Drivers → Face Driver → “Face” Driver Configurator → General Settings section (Figure 54)

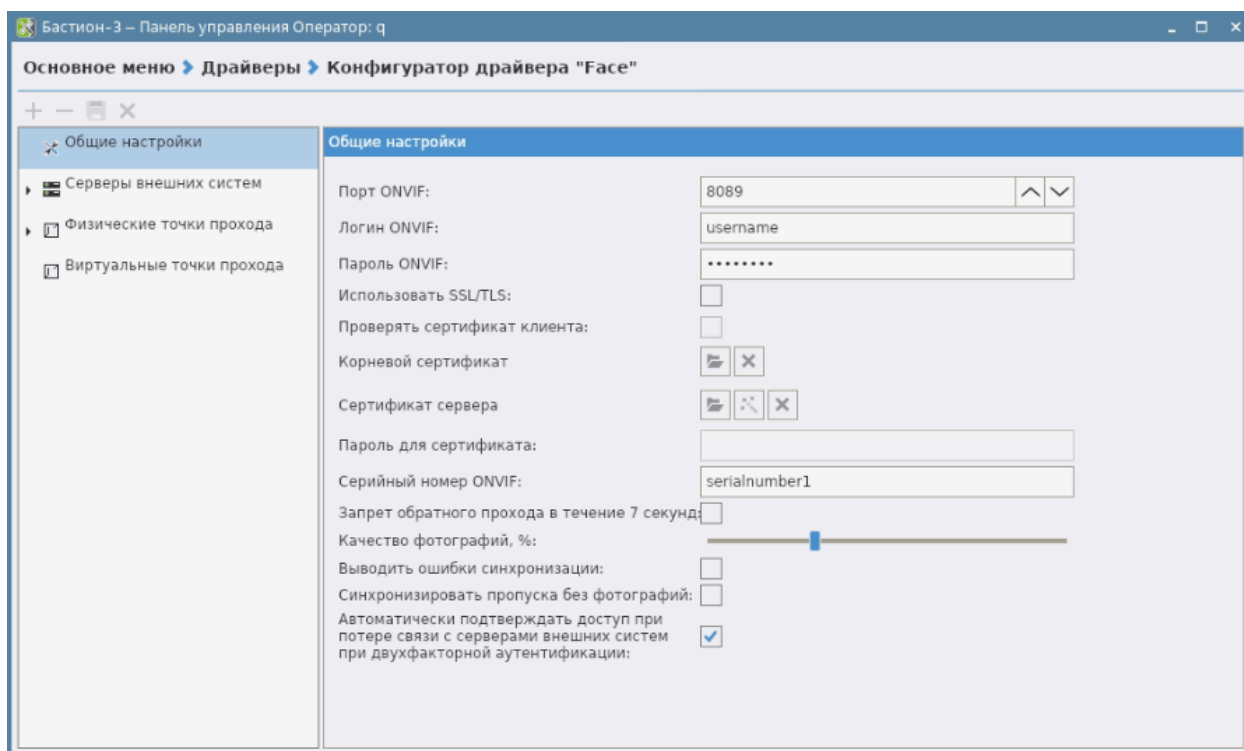


Figure 54. Face Driver Configuration

3. Set the ONVIF port, login, and password.
4. Go to the "External System Servers" section and add a new server by clicking "+".
5. In the new server settings, enter the Access address in the "host:port" format in the "person profile management service" and "event service" address fields, set the login and password for both services (Figure 55)

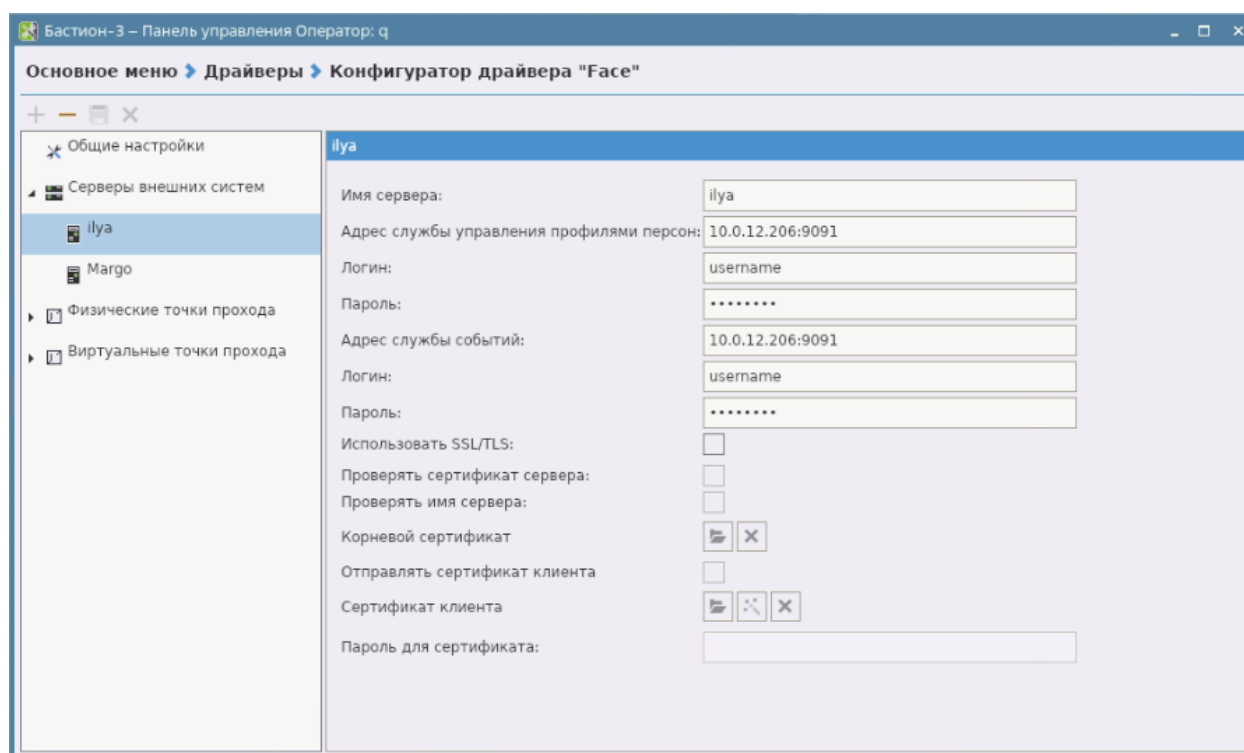


Figure 55. External systems server configuration

6. Go to the “Physical access points” section and add a new access point by clicking “+”.
7. Select the Door N R{N} access point.
8. In the “Description” field, enter the name of the camera that works with this access point.

The access point description must match the device name in Access.

9. Select the “Access in identification mode” operating mode.

When changing the access point mode in the ACS, you must restart the Bastion service in Access (Figure 56)

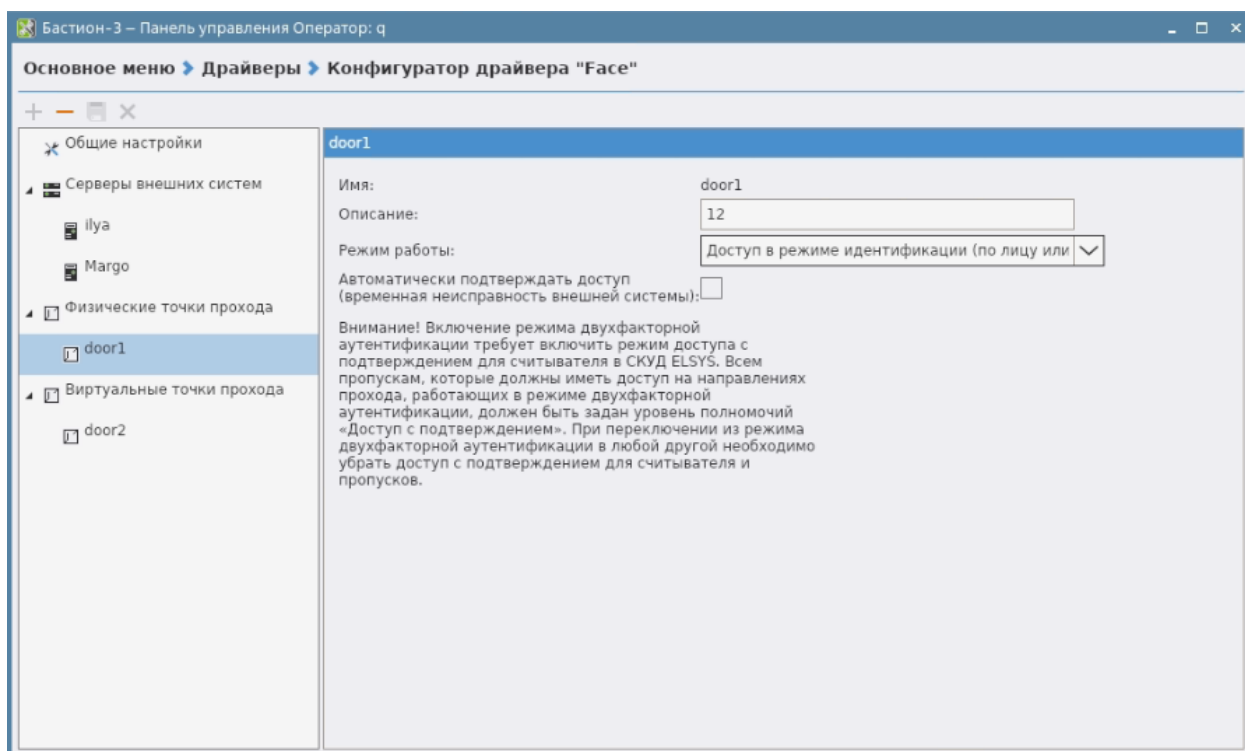


Figure 56. Access point configuration

10. Save the changes by clicking the floppy disk icon.
11. Configure pass management: Bastion 3 → Access Bureau.
12. Create a new pass request. Go to Requests → Click “+” on the toolbar (Figure 57).

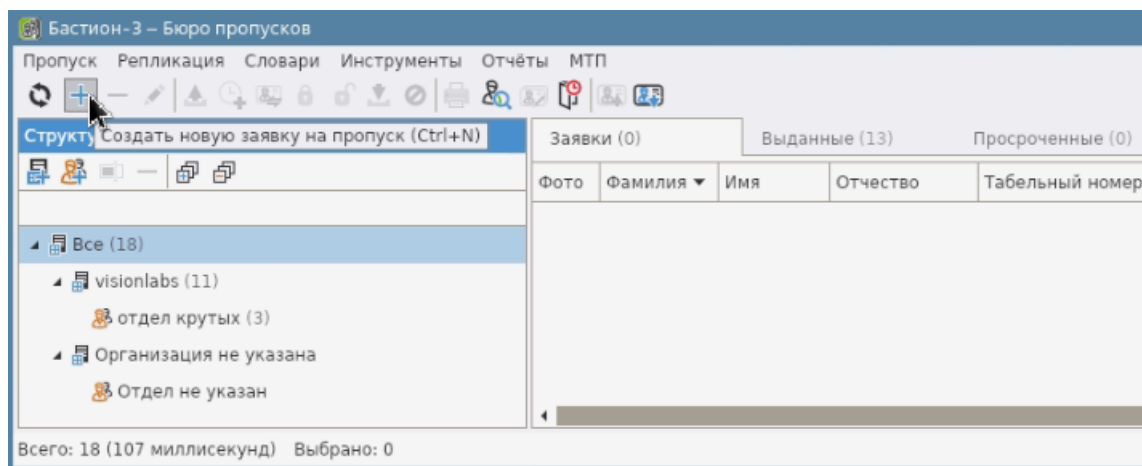


Figure 57. Create a pass request

13. Fill in the required fields and click OK (Figure 58)

Свойства пропуска

Персона | Пропуск | Уровень доступа | Профили | Реквизиты | Материальные пропуски ▾

Фамилия: Цискаридзе

Имя: Николай

Отчество: Николаевич

Организация: Все

Место работы: Все

Табельный номер:

Должность: <Не указано>

Комментарий:

Персона создана: 24.01.2025 11:31:19

OK Отмена

Figure 58. Fill out a pass request

14. Issue passes. Go to Requests → Select the target request → Click “Issue pass” on the toolbar (Figure 59)

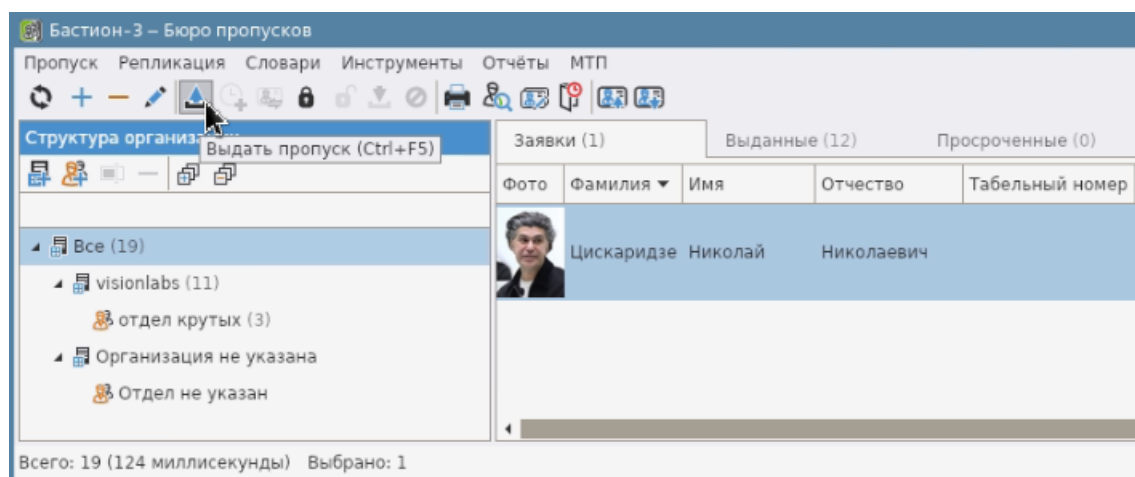


Figure 59. Issue passes

15. Check “Issue a new access card” → Generate a random code → Issue (Figure 60)

Выдача пропуска

Цискаридзе Николай Николаевич

☒ Выдать новую карту доступа

☒ HEX: 0000001BD841

☐ DEC: 0001824833

☐ Серия/номер: 027 55361

Сгенерировать случайный код

☐ Выдать существующую карту доступа

Код карты (HEX)	Код карты (DEC)	Код карты (серия/номер)
-----------------	-----------------	-------------------------

☐ Печать после выдачи

Выдать **Отмена**

Figure 60. Issue of passes

Issued passes are displayed on the “Issue” tab.

16. Editing a pass. Go to the Issued tab → Required pass → Pass properties.

10.3.1. Setting up a two-factor Bastion access point

1. Open the Bastion-3 ACS software — Control Panel.
2. Go to the Drivers → Face Driver → “Face” Driver Configurator → Physical Passage Points section and select/add a passage point.
3. In the “Description” field, enter the name of the camera that works with this passage point.

The passage point description must match the device name in Access.

4. Select the “Access in two-factor authentication mode” operating mode.

When changing the mode at the passage point in the ACS, you must restart the Bastion service in Access.

5. Save the changes by clicking the floppy disk icon (Figure 61)

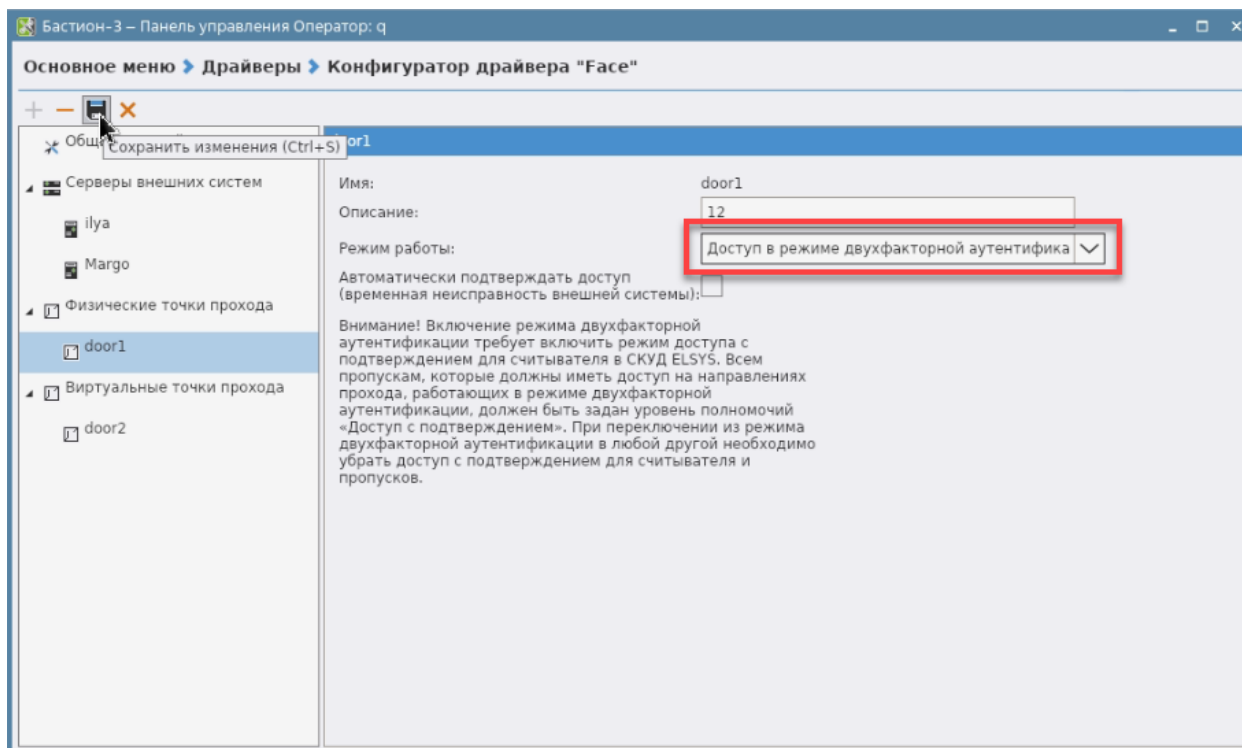


Figure 61. Passage point configuration for 2fa

6. Open the Control Panel and go to the Drivers → Personnel settings profiles section (Figure 62)

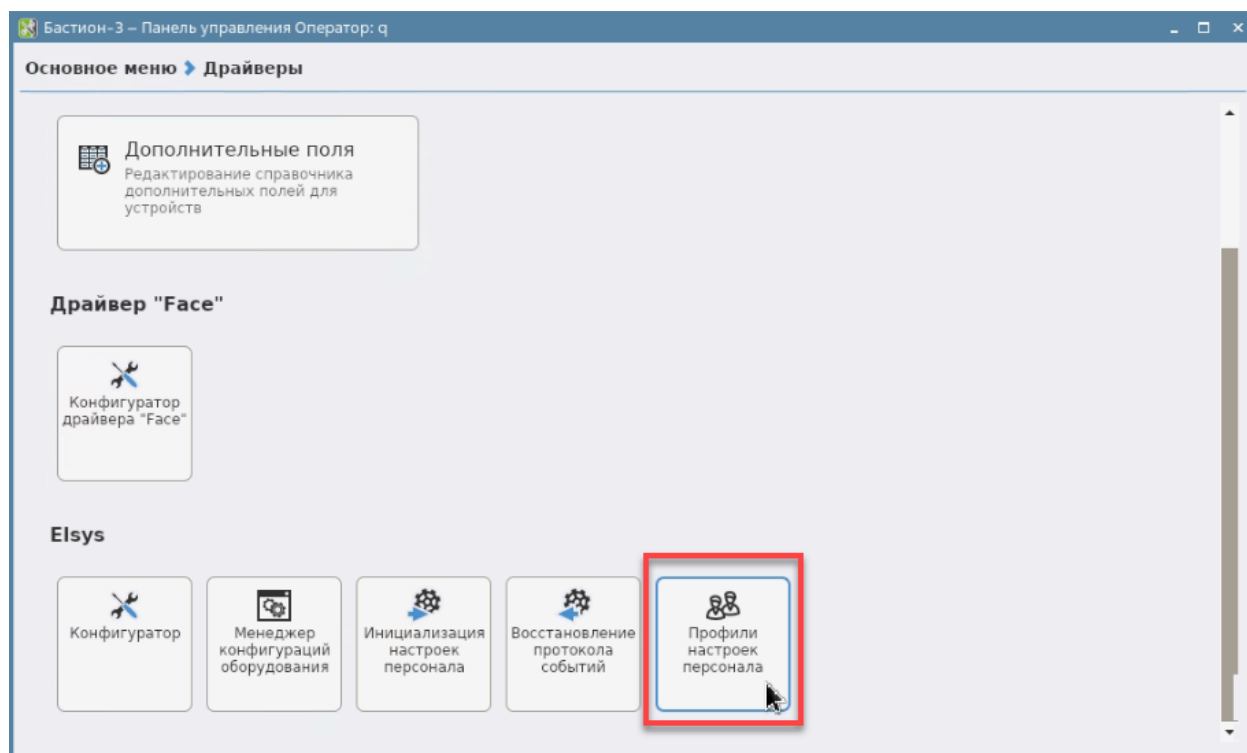


Figure 62. Staff settings profiles

7. Select a profile → Permissions and enable the “Access with confirmation” function.
8. Save the changes by clicking the floppy disk icon (Figure 63)

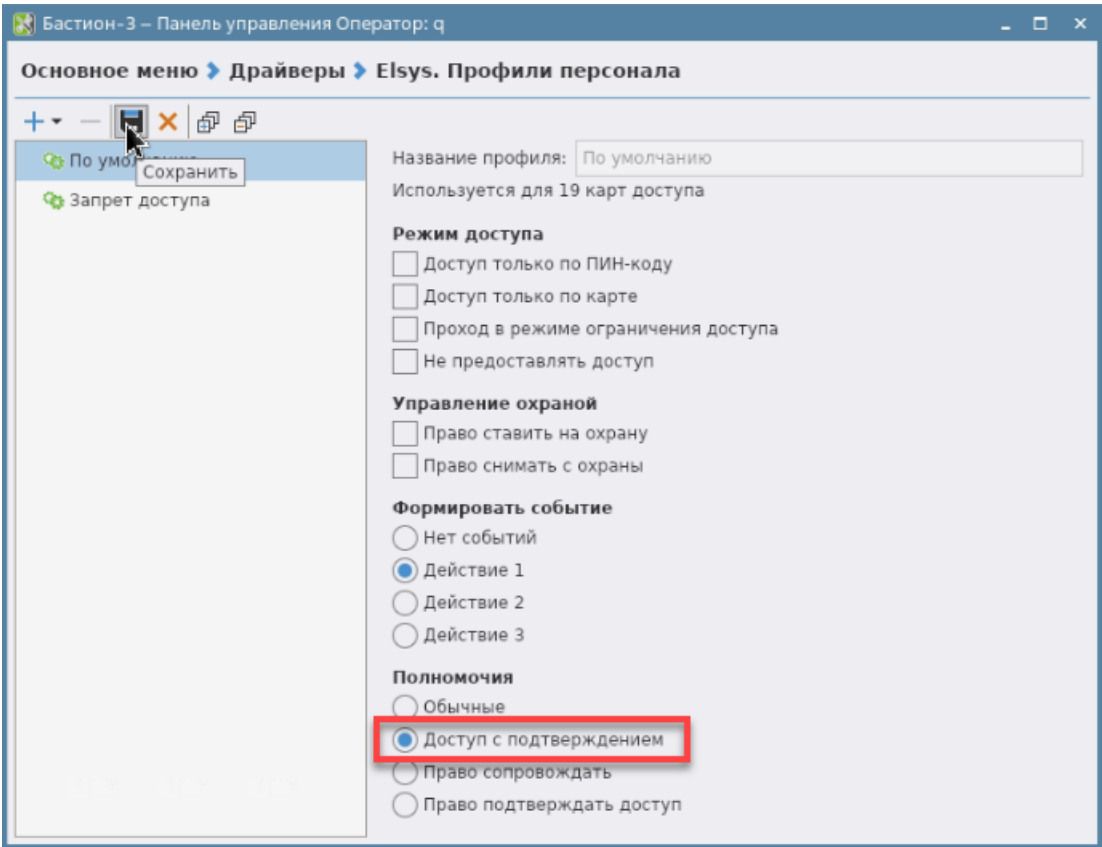


Figure 63. Enabling access with confirmation

9. In the Access UI, go to the “Services” tab and click the Bastion component restart button. In the Bastion component info, check the “enabled_2fa” setting of the access point that you edited in the previous step.

10.4. Methods of interaction with Bastion

Access acts as a server and a client (Table 42).

ONVIF methods are sent to Access at the POST /vl-access/webhook/service/onvif/{component_id} endpoint.

Table 42. Bastion methods

Task	Method	Description
Get access points	POST /onvif/accesscontrol	Request to ACS. Getting access point (controller) IDs for manual matching of cameras/terminals and access points
Get a list of ONVIF services	POST /onvif/device_service	Getting a list of component_id ONVIF Access services for connection

Task	Method	Description
Create user	CreateCredential	ONVIF method
Update User	ModifyCredential	ONVIF Method
Delete User	DeleteCredential	ONVIF Method
Create Subscription	CreatePullPoint Subscription	ONVIF Method. Subscribe to Events.
Get Detection Events	PullMessages	Get employee detection events. The request is sent every 10 seconds and waits 10 seconds until a frame appears.

10.5. Bastion Interaction Process Diagrams

10.5.1. Connecting the Bastion service and replicating employees

Sequence diagram (Figure 64).

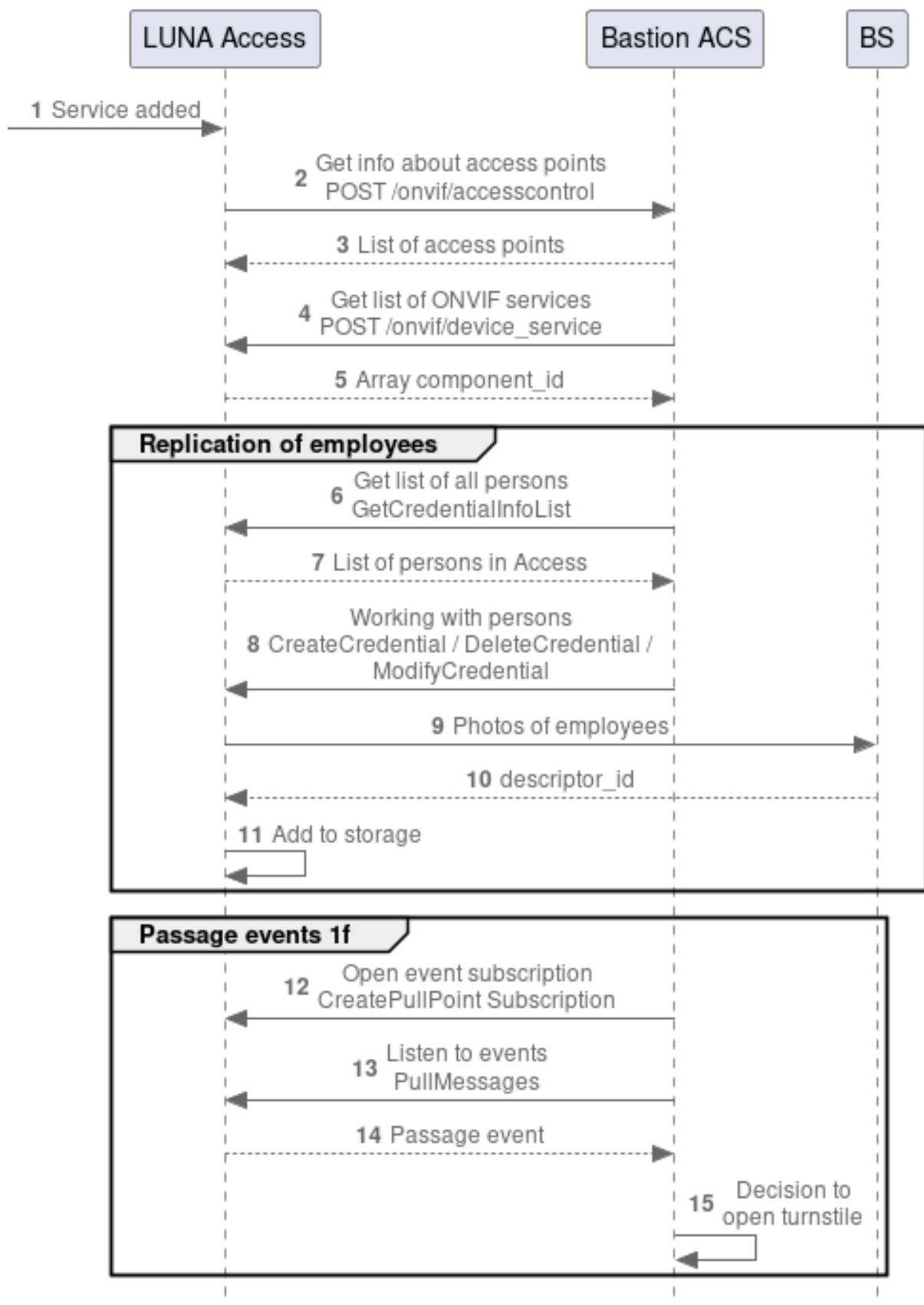


Figure 64. Process diagram for connecting the ACS

Connecting the service

1. The user added the Bastion service to Access.
2. Access sends a request to the ACS to obtain access points. The obtained access points are displayed in the info field of the service properties. The request is used to check the availability of the ACS.
3. The ACS returns access points.
4. The ACS sends a request to Access to obtain a list of Access services that support the ONVIF protocol.
5. Access returns the component_id of the ONVIF services.

Employee replication

6. ACS sends a request to Access to get a list of all persons.
7. Access returns a list of persons.
8. The ACS sends a POST request to Access at /vl-access/webhook/service/onvif/{component_id} CreateCredential (or DeleteCredential, ModifyCredential) to manage employees in the Access storage.
9. Access sends a request with employee photos to the BS to extract descriptor_id (face_id).
10. BS returns descriptor_id.
11. Access saves information on each employee to local storage.

Events with 1 factor

12. ACS sends a request to Access to open a subscription to receive events (the best shots of a person at the terminal).
13. The ACS sends a POST /vl-access/webhook/service/onvif/{component_id} PullMessages request every 10 seconds to wait for a pass event.
14. Access returns the pass event to the ACS.
15. The ACS makes a decision to open the terminal.

10.5.2. Processing Bastion events with 2 factors

Sequence diagram (Figure 65).

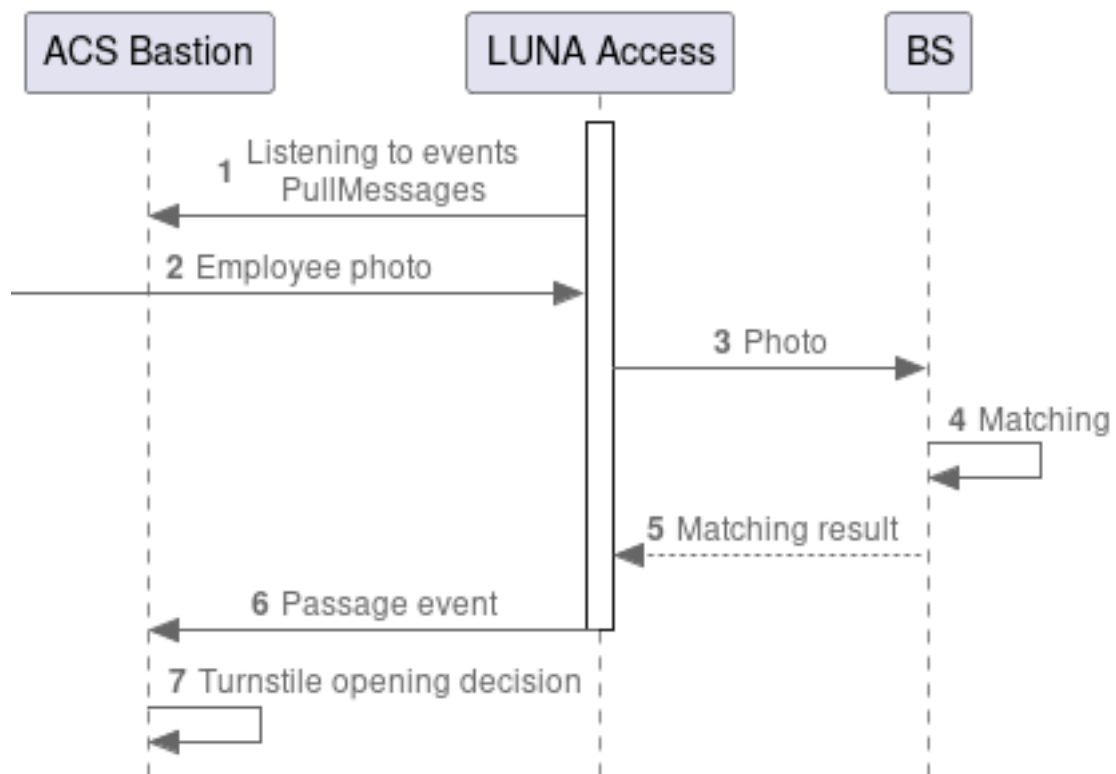


Figure 65. Process diagram with 2 factors

1. Access sends a PullMessages request to the ACS every 10 seconds to wait for a passage event.
2. Access receives the best photo of the employee at the terminal.
3. Access sends a photo of the employee to the Biometric System.
4. The BS compares the photos from the terminal and the one saved in the database.
5. The BS returns a decision to grant access to Access.
6. Access returns a passage event to the ACS.
7. The ACS decides to open the terminal.

11. Bolid ACS

Hardware and software integration required for communication between the LP5/CBS and the Bolid ACS software to ensure control of the connected device (C-2000 series devices or other devices compatible with the Bolid software).

Supports Bolid version 1.20.3, Orion Pro integration module version 1.4.

Information interaction is provided through the Orion Pro automated workplace software.

The Orion PRO licensed integration module must be installed and launched.

The integration module is a SOAP web service accessed via the HTTP/HTTPS protocols. The description of the web service complies with the WSDL version 2.0 specification.

The service runs under Windows 7/8/8.1/10 (32 bit or 64 bit).

11.1. Supported integration options for Bolid ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

User data is transferred from ACS to LP5 using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

For the synchronization/replication settings, see the service settings.

Each integration with LP5 (Table 43) uses the [Luna](#) service.

If the terminal does not have data output means (for example, a screen), the [SendToDevice](#) pipeline is not required.

Table 43. LP5 integration options

Service	Device	Pipeline
1f		
Bolid + GateController / PusrController	Beward	MatchByPhoto + SendToDevice + SendToController
	BioSmart	MatchByPhoto + SendToDevice + SendToController
	Dahua	MatchByPhoto + SendToController
	Dahua Thermo	MatchByPhoto + SendToController

Service	Device	Pipeline
	Fortuna315	MatchByPhoto + SendToController
	HikvisionCamera	MatchByPhoto + SendToController
	HikvisionCamera Thermo	MatchByPhoto + SendToController
	HikvisionTerminal Thermo	MatchByPhoto + SendToDevice + SendToController
	LunaFast4A1	MatchByPhoto + Custom2FA
	Panda	MatchByPhoto + SendToController
	UniUbi	MatchByPhoto + SendToDevice + SendToController
	VKVision02	LunaStreams + MatchByPhoto + SendToDevice + SendToController
	R20Face	MatchByPhoto + SendToDevice + SendCardToR20Face
2f		
Bolid + GateController / PusrController	LunaFast4A1	Custom2FA + MatchByPhoto + SendToDevice

Each integration with CBS (Table 44) uses the CBS service.

Table 44. CBS integration options

Service	Device	Pipeline
CbsMts + Bolid + GateController / PusrController	Beward	MatchByPhoto + SendToController + SendToDevice
	Dahua	MatchByPhoto + SendToController
	HikvisionCamera	MatchByPhoto + SendToController
	LunaFast4A1	MatchByPhoto + SendToController + SendToDevice
	UniUbi	MatchByPhoto + SendToController + SendToDevice

Service	Device	Pipeline
	R20Face	MatchByPhoto + SendCardToR20Face + SendToDevice

11.2. Standard integration using Bolid

When integrating with Bolid, standard Access components (Figure 66) and (Table 45) are used.

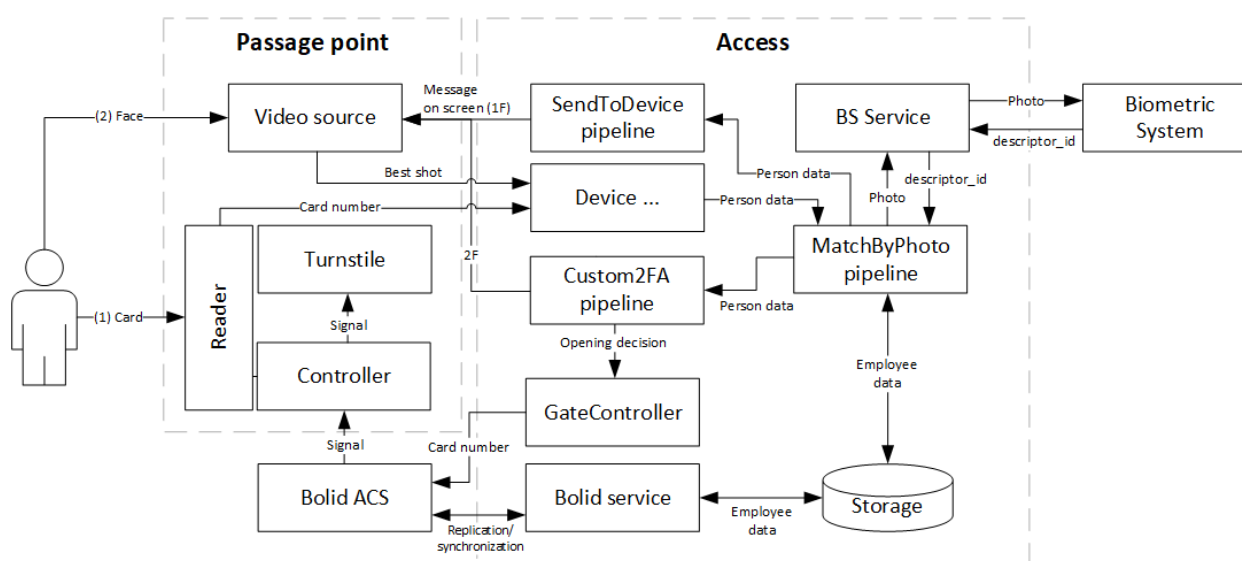


Figure 66. Component diagram for integration with Bolid

When using 1f integration (without a card) components, transferring the card number to Access is not required.

Table 45. Integration description

Component	Description
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Reader	A device for receiving access card data.
Controller	A passage point control board.
Turnstile	A barrier device for access control

Component	Description
Bolid ACS	Central software for working with Bolid. Stores employee data and makes a decision on granting access.
Bolid Service	Access component for exchanging data with ACS
GateController	Access component for interacting with the ACS controller.
BS Service	Access component for interaction with the BS: for LP5 it is Luna , for CBS - the corresponding CBS service.
Video data source	A device for extracting a frame of a person's face. It can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream.
Device ...	An Access component for receiving data from a video data source. Selected based on the device used.
Custom2FA Pipeline	Access Component implementing the logic when working in 2f mode.
MatchByPhoto Pipeline	Access Component for interaction with CBS. When working with a biometric terminal, it is necessary to additionally connect the SendToDevice pipeline
SendToController Pipeline	Access Component for interaction with CBS

11.3. Setting Bolid ACS

11.3.1. Preparatory actions with Orion Pro software

To launch and configure Bolid, you need to perform preparatory steps with the Orion Pro software:

1. Launch the Orion Pro Central Server application.
2. Launch the Orion Shell application.
3. Launch the ADB module on the Orion Shell panel (Figure 67):

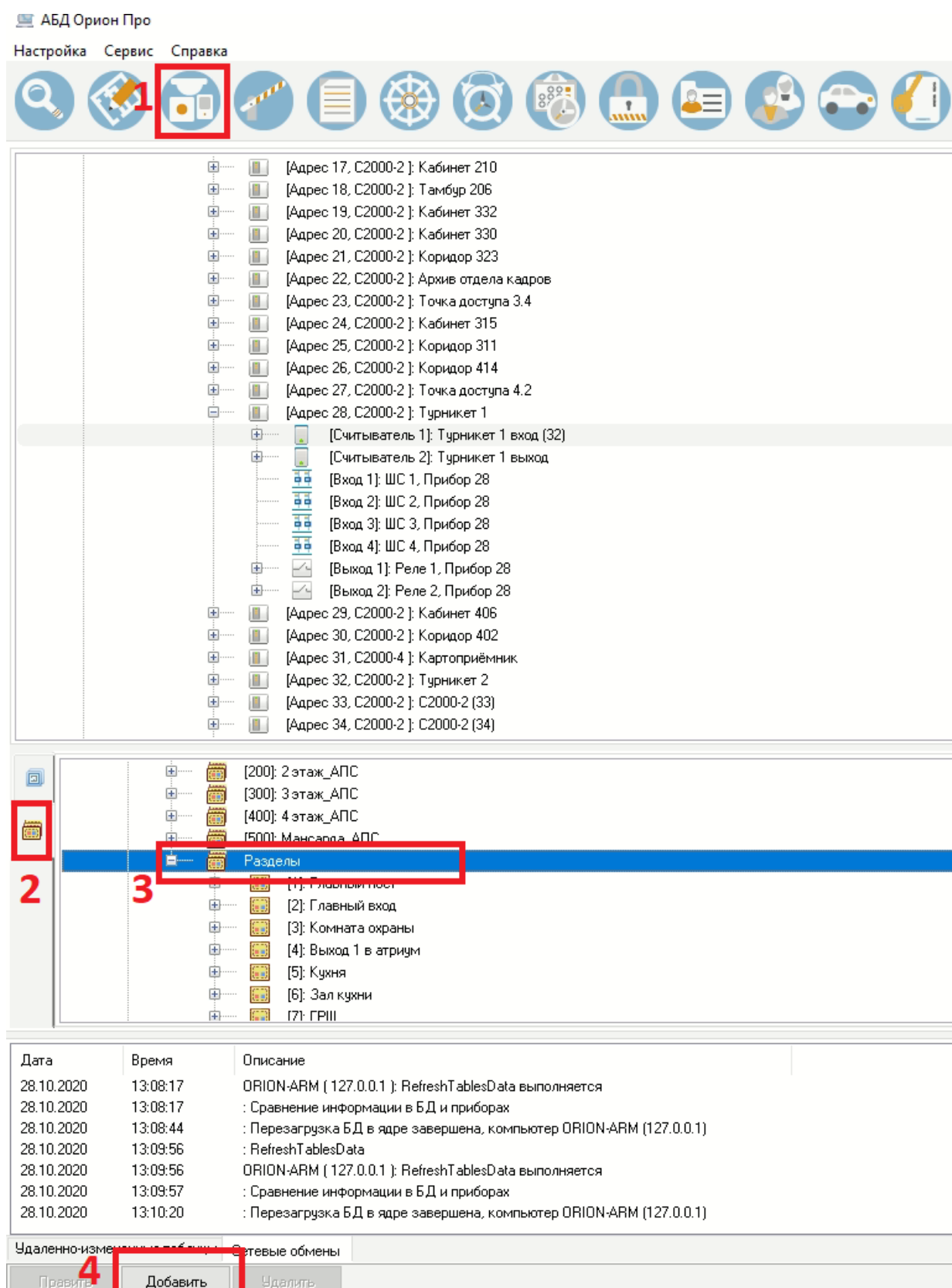


Figure 67. Launching the ADB on the Orion Shell panel

4. Launch the Orion Pro Integration Module.
5. Launch the Operational Tasks (OT) on the Orion Shell panel if they do not start automatically.

11.3.2. Adding an employee in Orion Pro

1. Add a new employee. Fill in the required fields (Figure 68) according to the rules for creating employees at the facility:
2. Go to the Employees section
3. Click the “add” button
4. Fill in the required employee fields

Select the “Administrator” status, or another department with employees who have full access to the system.

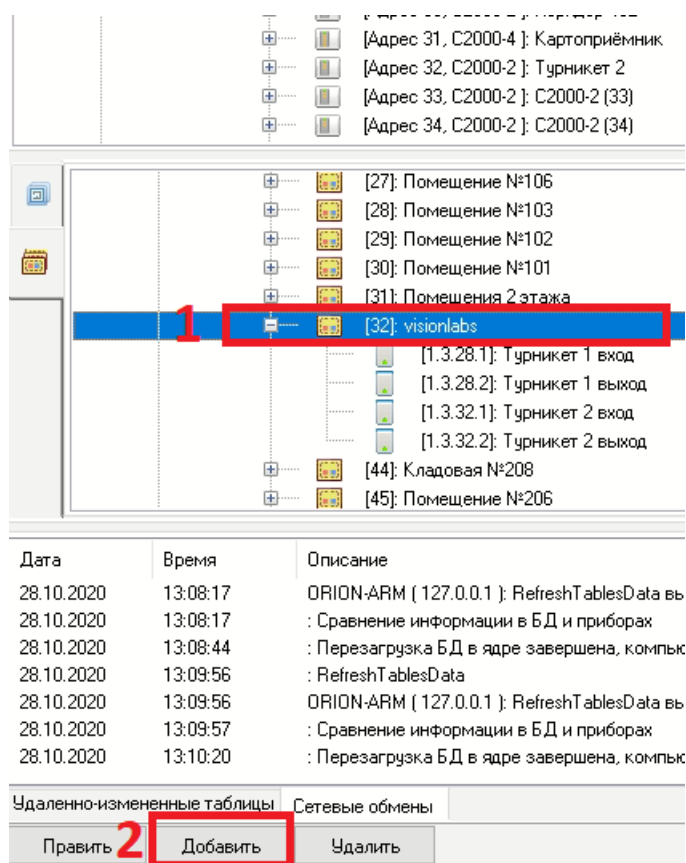


Figure 68. Adding a new employee

5. Add the “Maximum” access level to the new user and set a password (Figure 69).
6. Go to the Access section;

7. Click Add;
8. Select the desired employee, enter the password
9. Select the Maximum access level

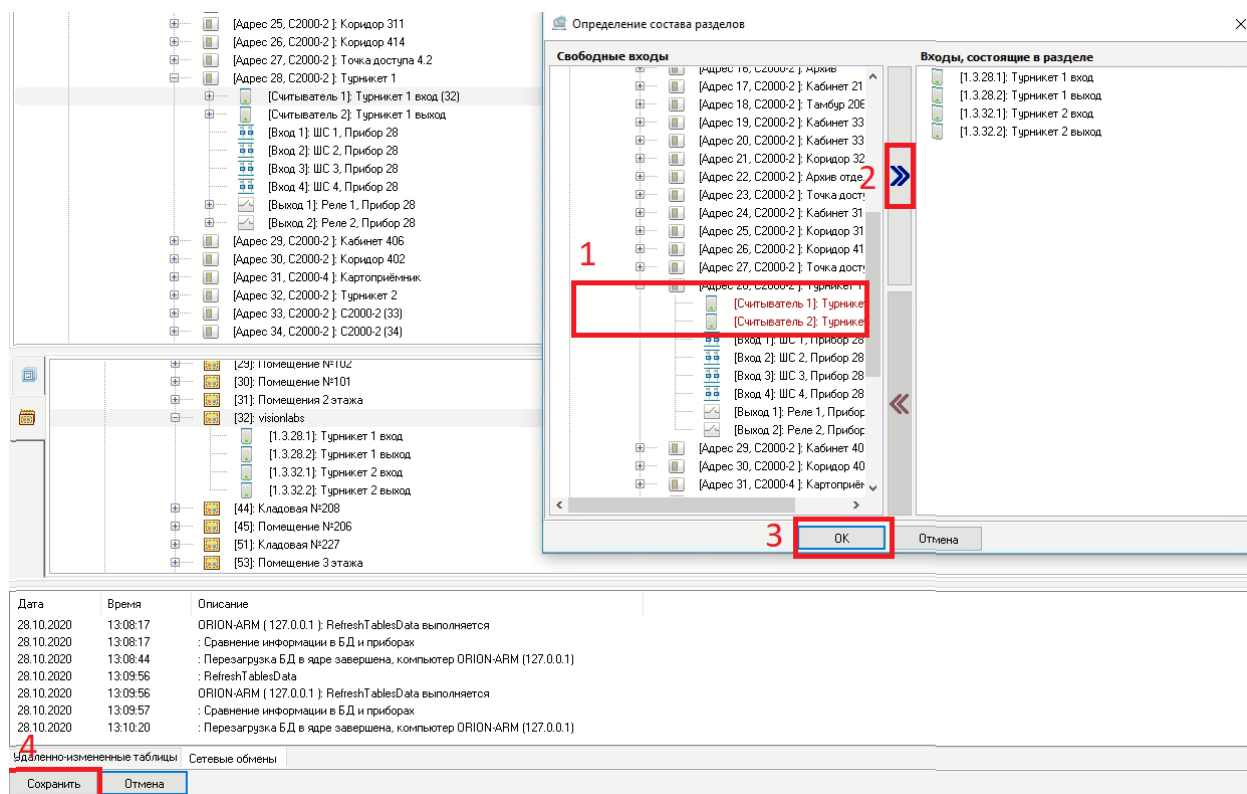


Figure 69. Setting an access level for an employee

11.3.3. Adding devices to Orion Pro

1. Add a new section (Figure 70):
2. Select the “System structure” tab;
3. Select “sections”;
4. Select all “Sections”;
5. Add a new section with standard parameters and name it.

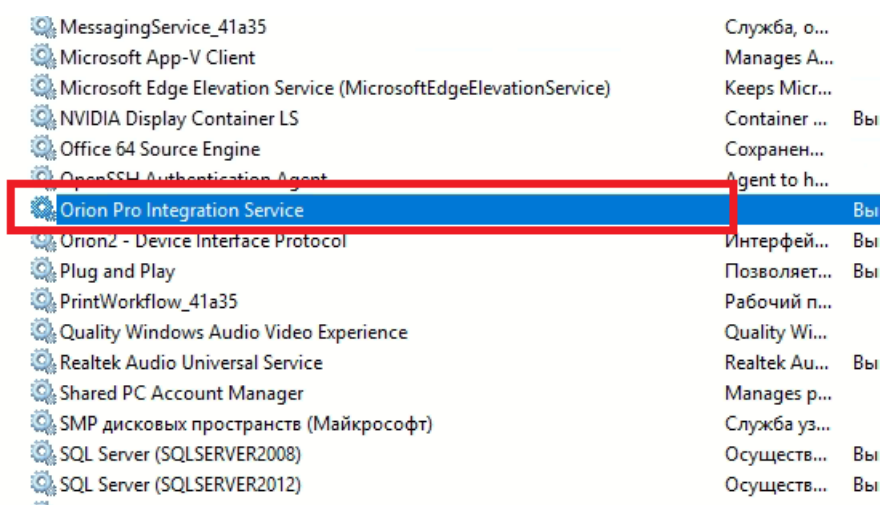


Figure 70. Setting an access level for an employee

6. Link devices to the newly created section (Figure 71):
7. Select a section.
8. Click the “Add” button.

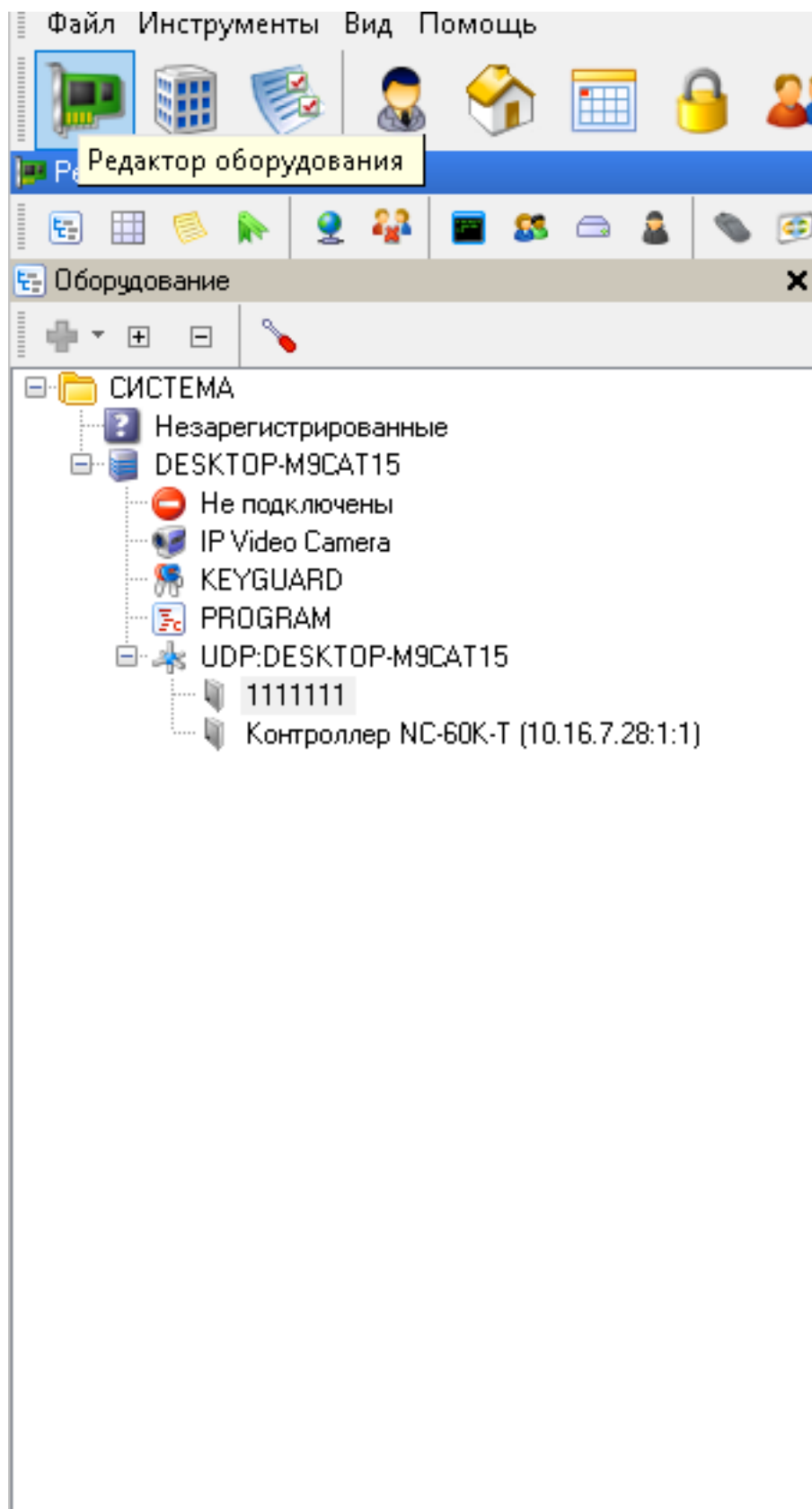


Figure 71. Linking devices to a section

9. Go to the list of devices and select the required ones (Figure 72):
10. Select it and click the [>>] button to move it to the active field;
11. Confirm the changes by clicking [OK];
12. Click the “Save” button.

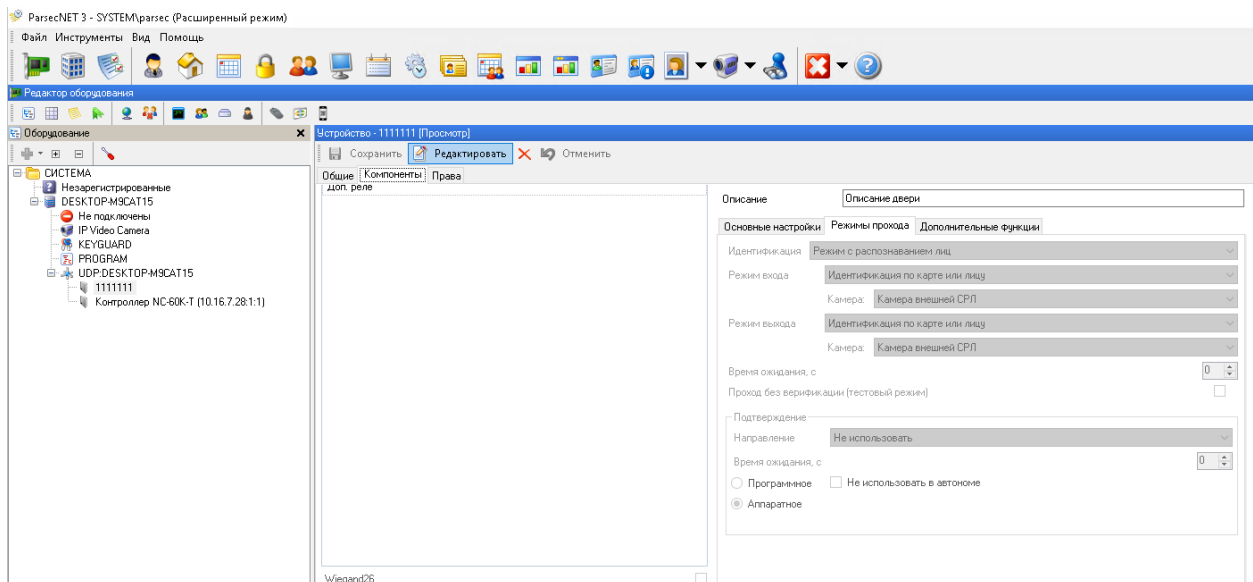


Figure 72. Linking devices to the partition

13. Edit the orion.ini file in the folder with the installed Orion Pro application (located by default: C:\BOLID\ARM_ORION_PRO1_20_3), adding the following parameters (if they are missing):

```
[Checkerdb]
Remarks=1
timechecker=5
Logon=1
RemoteCmd=1
CmdOn=1
[ChangeDB]
on=1
```

14. Restart all Orion Pro applications.

11.3.4. Configuring the “ORION PRO INTEGRATION MODULE” application

To configure the “ORION PRO INTEGRATION MODULE” application, follow these steps:

1. Download the official distribution kit of the “ORION PRO INTEGRATION MODULE” application (link).

- 2. Run the installation. After the installation is complete, launch the module, check the database connection settings. If everything works correctly, close the module.
- 3. Install the module to run as a service. To do this, run the command in the terminal as administrator in the folder with the installed module (for example: C : \BOLID\IntegrServ):

```
IntegrServ.exe /INSTALL
```

- 4. In the system control panel, find the installed service and run it by clicking the right mouse button and selecting “Start” (Figure 73).

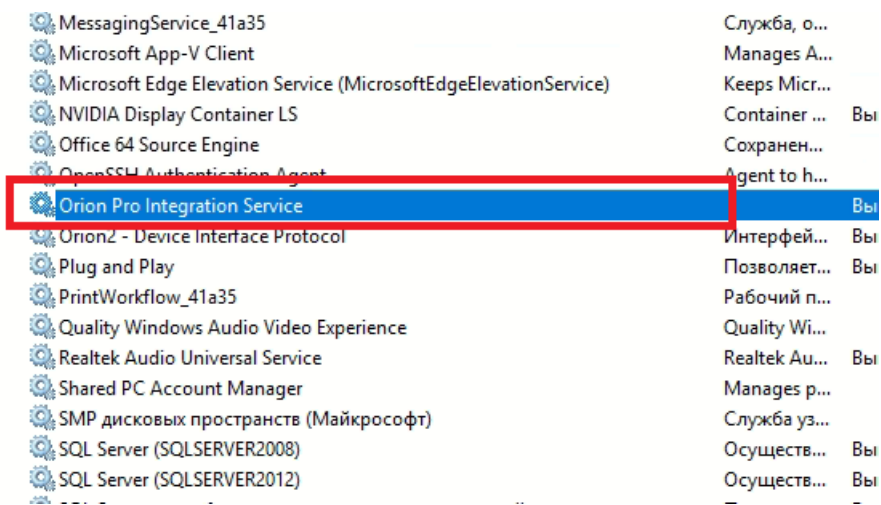


Figure 73. System control panel

11.4. Methods of interaction with Bolid

An API is used to exchange data with the ACS (Table 46).

Table 46. Bolid methods

Task	Operation	Description
Get information about ACS	GetServiceId	Get ACS version to check compatibility and display in UI. Check ACS availability (every minute).
Get authorization status	GetDbChangeEngineStatus	Check the status of correct login/password entry for an account in ACS.
Log in	GetLoginTo	Authorization of Access in ACS. Authorization occurs when adding a service and before token expiration (10 minutes)

Task	Operation	Description
Extend token	ExtendTokenExpiration	When executing requests, the token lifetime is checked. The request is executed if the token lifetime is about to expire
Get employees	GetPersons	Replication and synchronization of employees (person_id, full name, status, photo, date and time of the last update) from the ACS to the local storage. Iteratively by 500 rows.
Get employee information	GetPersonById	Getting employee data from the ACS (person_id, full name, status, photo, date and time of the last update)
Get card number	GetKeys	Getting PROXIMITY cards for all employees. The most recent ones are selected for each employee
Get events	GetDbChanges	Getting employee events (adding, changing or deleting) every 5 seconds.

11.5. Bolid Interaction Process Diagrams

11.5.1. Bolid Service Connection

Sequence diagram (Figure 74).

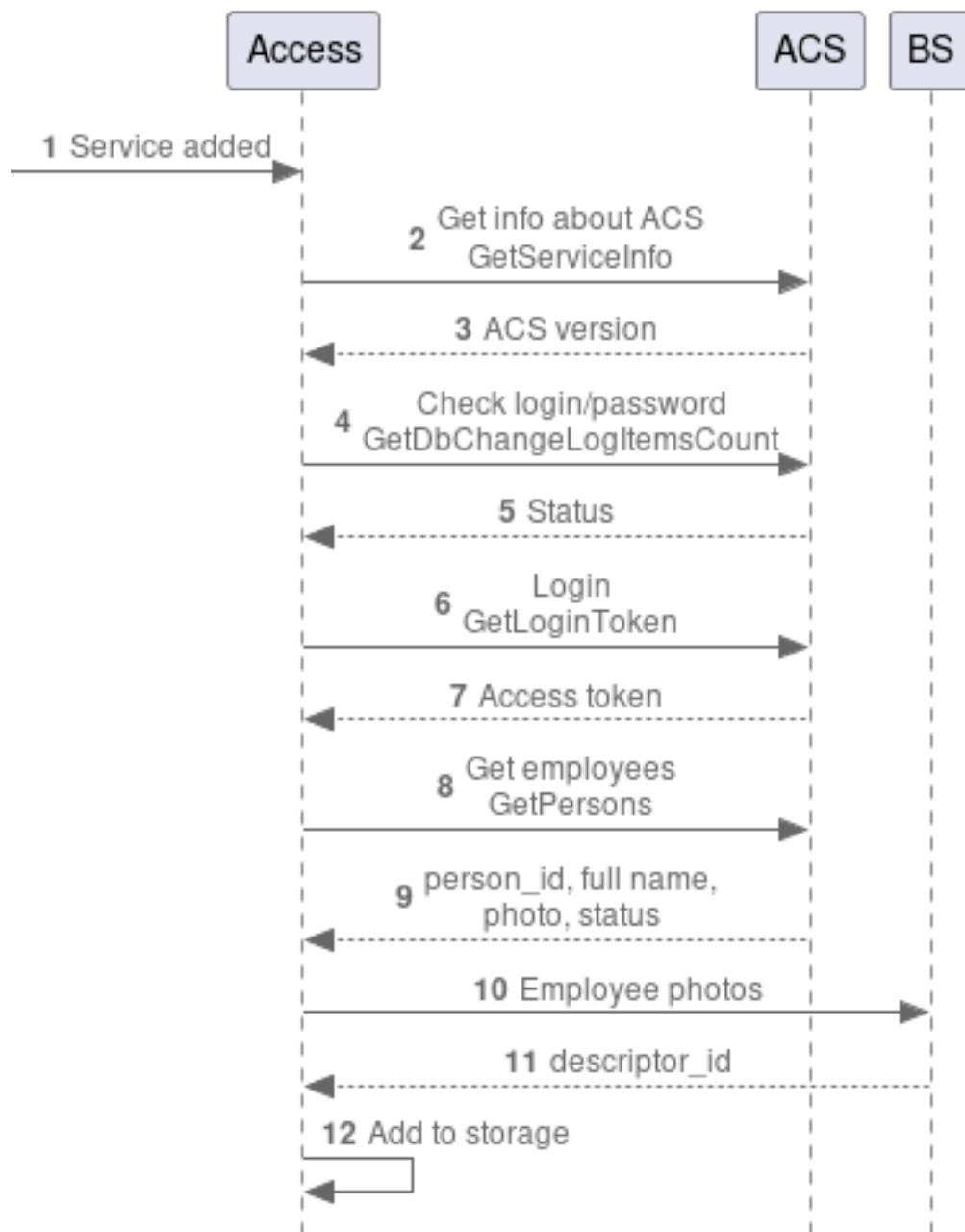


Figure 74. Process Diagram for ACS Connection

1. The user added the Bolid service to Access.
2. Access sends a request to obtain information about the ACS.
3. The ACS returns information. Access checks the availability of the ACS and uses the ACS version to check compatibility and user information in the UI.
4. Access sends a request for the correctness of the login/password pair from the account in the ACS.
5. The ACS returns the account activity status. If the record is active, then work continues.

6. Access sends an authorization request to the ACS.
7. The ACS returns a token for authorization. The token has a lifetime, after which Access re-performs authorization.
8. Access sends a request to obtain information about employees to replicate data to the local storage.
9. The ACS returns person_id, full name, status, photo, date and time of the last update.
10. Access sends a request with employee photos to the BS to retrieve descriptor_id (face_id).
11. The BS returns descriptor_id.
12. Access saves information on each employee in local storage.

11.5.2. Event processing with 1 factor

Sequence diagram (Figure 75).

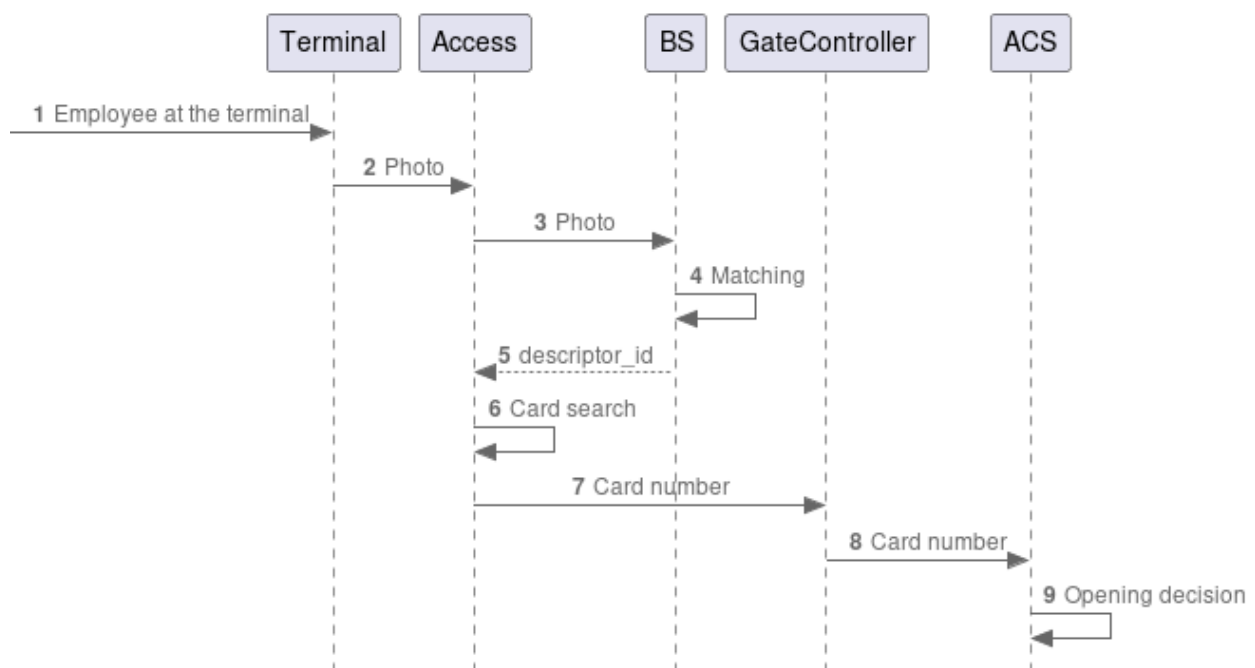


Figure 75. Process diagram with 1 factor

1. Employee at the biometric terminal at the checkpoint.
2. The terminal sends the best shot of the employee to Access.
3. Access sends a photo of the employee to the Biometric System (BS).
4. BS compares the photo from the terminal and the one saved in the database.
5. BS returns descriptor_id to Access.

6. Access looks for the employee's card number by correlating descriptor_id and person_id.
7. Access sends the card number to GateController.
8. GateController sends the card number to ACS.
9. ACS makes a decision to allow the employee through.

11.5.3. Event processing with 2 factors

Sequence diagram (Figure 76).

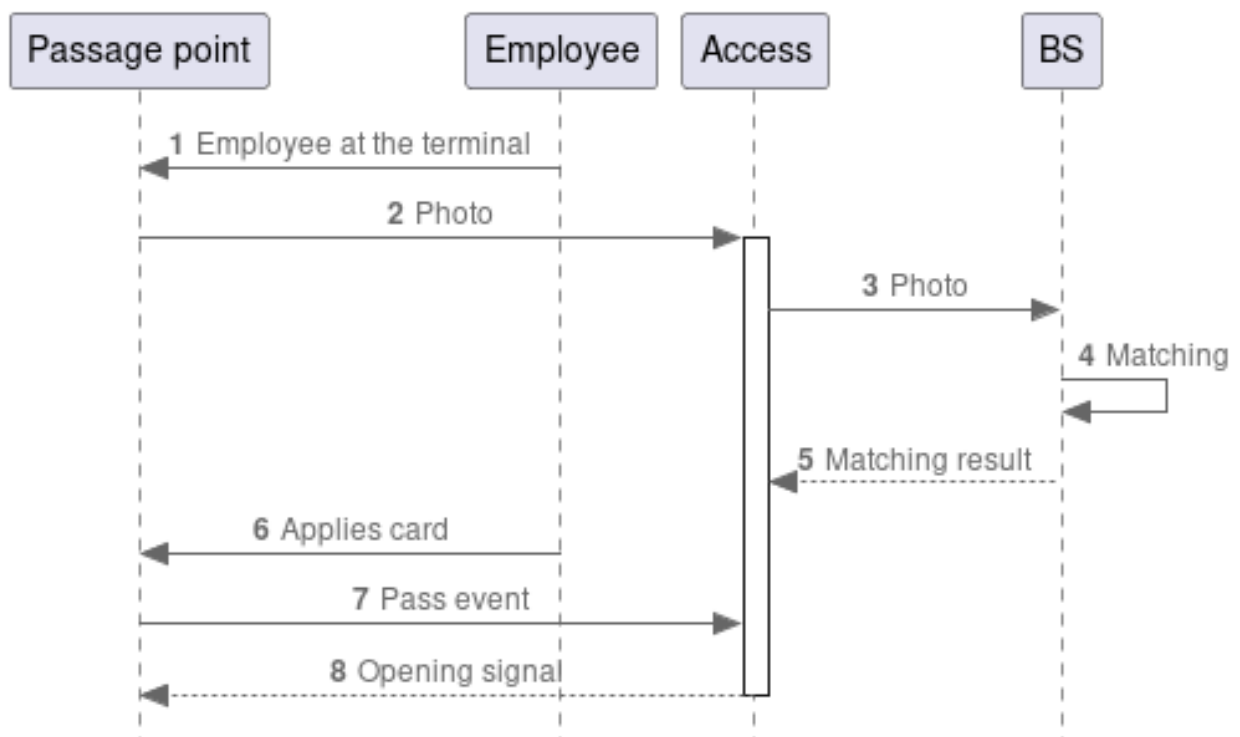


Figure 76. Process diagram with 2 factors

1. Employee at the biometric terminal at the checkpoint.
2. The terminal sends the best shot of the employee to Access.
3. Access sends the employee's photo to the Biometric System.
4. The BS compares the photo from the terminal and the one saved in the database.
5. The BS returns the descriptor_id to Access.
6. Access looks for the employee's card number by correlating the descriptor_id and person_id.
7. The employee applies the card (the card use subprocess does not depend on photo processing, but, as a rule, the photo arrives first).

8. The ACS sends the card number to Access.
9. Access compares the card numbers received in steps 6 and 8.
10. Access sends the card number to the GateController.
11. GateController sends the card number to the ACS.
12. The ACS makes a decision to open.

11.6. Bolid FAQ

1. Why can't I add an employee photo larger than 100 kb?
 - In Orion Pro, the maximum permissible photo size may be reset to 100 kb. You need to go to ADB Orion Pro Settings > Settings > Employees > "Maximum size of employee photos, kb" and set it to 10240 kb (10 MB) or more.

12. Gate ACS

The integration module (sync.exe), launched in the Gate Server directory, detects changes in the database and sends changes to the ip:port specified in the .env settings file. VL Access accepts and processes these requests, and makes appropriate changes to the Luna list.

Supported versions: Gate Terminal 1.22.95, Gate Server 1.22.95

12.1. Supported integration options for Gate ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

Transfer of user data from the ACS to LP5 occurs using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

For the synchronization/replication settings, see the service settings.

Each integration with LP5 (Table 47) uses the [Luna](#) service.

If the terminal does not have data output facilities (e.g., a screen), the [SendToDevice](#) pipeline is not required.

Table 47. LP5 integration options

Service	Device	Pipeline
Gate + GateController / PusrController	Beward	LunaEventListener + SendThermalEventToLuna/SendToLuna
	BioSmart	LunaEventListener + SendToLuna
	Dahua	LunaEventListener + SendToLuna
	Dahua Thermo	LunaEventListener + SendThermalEventToLuna
	Fortuna315	LunaEventListener + SendThermalEventToLuna
	HikvisionCamera	LunaEventListener + SendToLuna
	HikvisionCamera Thermo	LunaEventListener + SendThermalEventToLuna
	HikvisionTerminal Thermo	LunaEventListener + SendThermalEventToLuna

Service	Device	Pipeline
	LunaFast4A1	LunaEventListener + SendToLuna
	Panda	LunaEventListener + SendThermalEventToLuna
	UniUbi	LunaEventListener + SendThermalEventToLuna / SendToLuna
	VKVision02	LunaEventListener
	R20Face	LunaEventListener + SendToLuna

13. Parsec ACS

Supports ParsecNET 3 ACS version: 3.11.629 39.

The service allows you to process requests from access control systems, such as:

- transfer the list of employees to the local person storage,
- adding/editing/deleting employees in the local person storage,
- receiving detection events from devices.

The service executes the following requests to the ACS:

- sending url address of ONVIF services;
- getting access point IDs.

When starting the service, access point identifiers are first requested and their names are generated.

The ACS queries Access for detections and generates a response containing the employee ID and the access point ID.

As soon as a valid face detection occurs, the service returns a response to the ACS.

To integrate with Parsec, you must specify the names of access points automatically generated by the service in the device settings. They are generated in the format “access point name - identifier”. For example: “Turnstile exit - 907EFA78-CB2F-4F46-b374-785c7f9901a5”

The resulting access point names must be inserted into the appropriate fields:

- When using internal Access devices (HikvisionTerminal, Panda ...), indicate in the “name” field.
- When using LunaStream, indicate in the “source” field.

13.1. Supported integration options for Parsec ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

Transfer of user data from the ACS to LP5 occurs using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

For the synchronization/replication settings, see the service settings.

Each integration with LP5 (Table 48) uses the [Luna](#) service.

If the terminal does not have data output facilities (e.g., a screen), the [SendToDevice](#) pipeline is not required.

Table 48. LP5 integration options

Service	Device	Pipeline
Parsec	Beward	SendToParsec + MatchByPhoto + SendToDevice
	BioSmart	SendToParsec + MatchByPhoto + SendToDevice
	Dahua	SendToParsec + MatchByPhoto
	Dahua Thermo	SendToParsec + MatchByPhoto
	Fortuna315	SendToParsec + MatchByPhoto
	HikvisionCamera	SendToParsec + MatchByPhoto
	HikvisionCamera Thermo	SendToParsec + MatchByPhoto
	HikvisionTerminal Thermo	SendToParsec + MatchByPhoto + SendToDevice
	LunaFast4A1	SendToParsec + MatchByPhoto
	Panda	SendToParsec + MatchByPhoto
	UniUbi	SendToParsec + MatchByPhoto + SendToDevice
	VKVision02	SendToParsec + MatchByPhoto + SendToDevice
	R20Face	SendToParsec + MatchByPhoto + SendToDevice

Each integration with CBS (Table 49) uses the CBS service.

Table 49. CBS integration options

Service	Device	Pipeline
CbsMts + Parsec	Beward	MatchByPhoto + SendToDevice + SendToParsec
	Dahua	MatchByPhoto + SendToParsec
	HikvisionCamera	MatchByPhoto + SendToParsec
	LunaFast4A1	MatchByPhoto + SendToDevice + SendToParsec
	UniUbi	MatchByPhoto + SendToDevice + SendToParsec

13.2. Standard integration using Parsec

When integrating with Parsec, standard Access components (Figure 77) and (Table 50) are used.

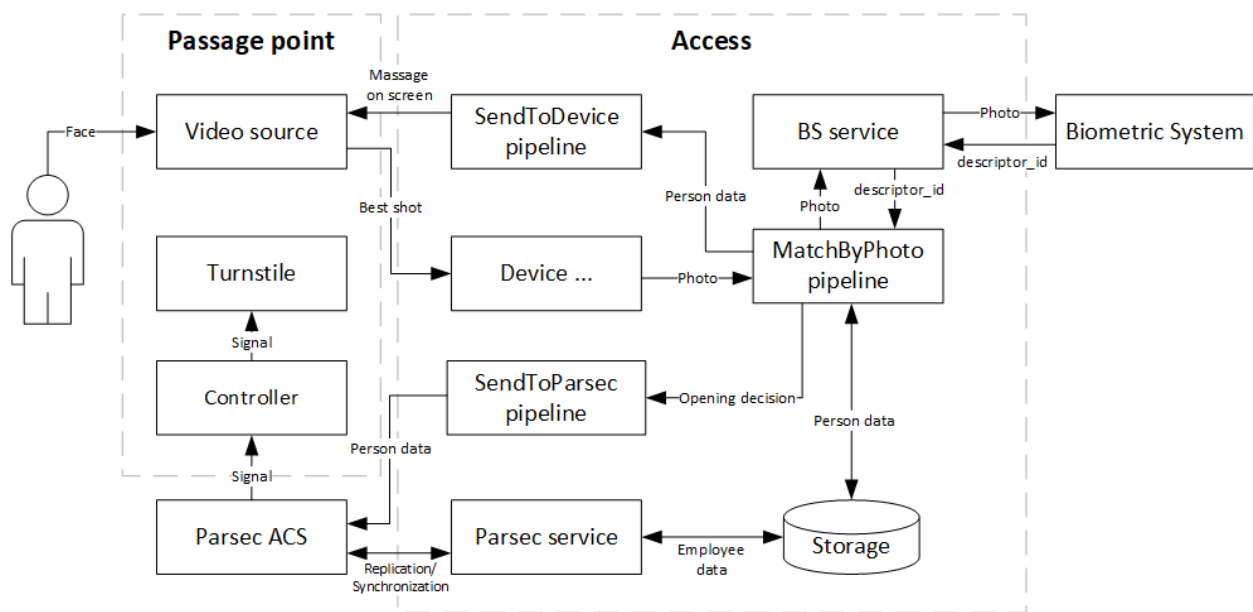


Figure 77. Integration Components Diagram

Table 50. Integration Description

Component	Description
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Controller	Passage point control board.
Turnstile	A barrier device for access control
Parsec ACS	Central software for working with Parsec. Stores employee data and makes decisions about granting access.
SendToParsec Pipeline	Access component for exchanging data with ACS
Parsec Service	Access component for processing information from the ACS
MatchByPhoto pipeline	Access component for interaction with the BS. When working with a biometric terminal (for displaying messages and photos on the screen), it is necessary to additionally connect the SendToDevice pipeline
SendToController pipeline	Access component for interaction with the CBS

Component	Description
Video source	A device for extracting a frame of a person's face. It can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via LunaStream.
Device ...	Access component for receiving data from a video data source. It is selected based on the device used.
Biometric system	A system for comparing a reference photo of a person and the best frame received from a video data source. It can be either Luna or CBS MTS .

13.3. Configuring Access and Parsec ACS integration

To launch and configure the Parsec ACS software, install Parsec.NET and run the Administration program and check the settings (Figure 78):

1. Make sure that the "Advanced Mode" is running (File→Advanced Mode).
2. Go to the "Equipment editor" section and make sure that the controllers are connected .

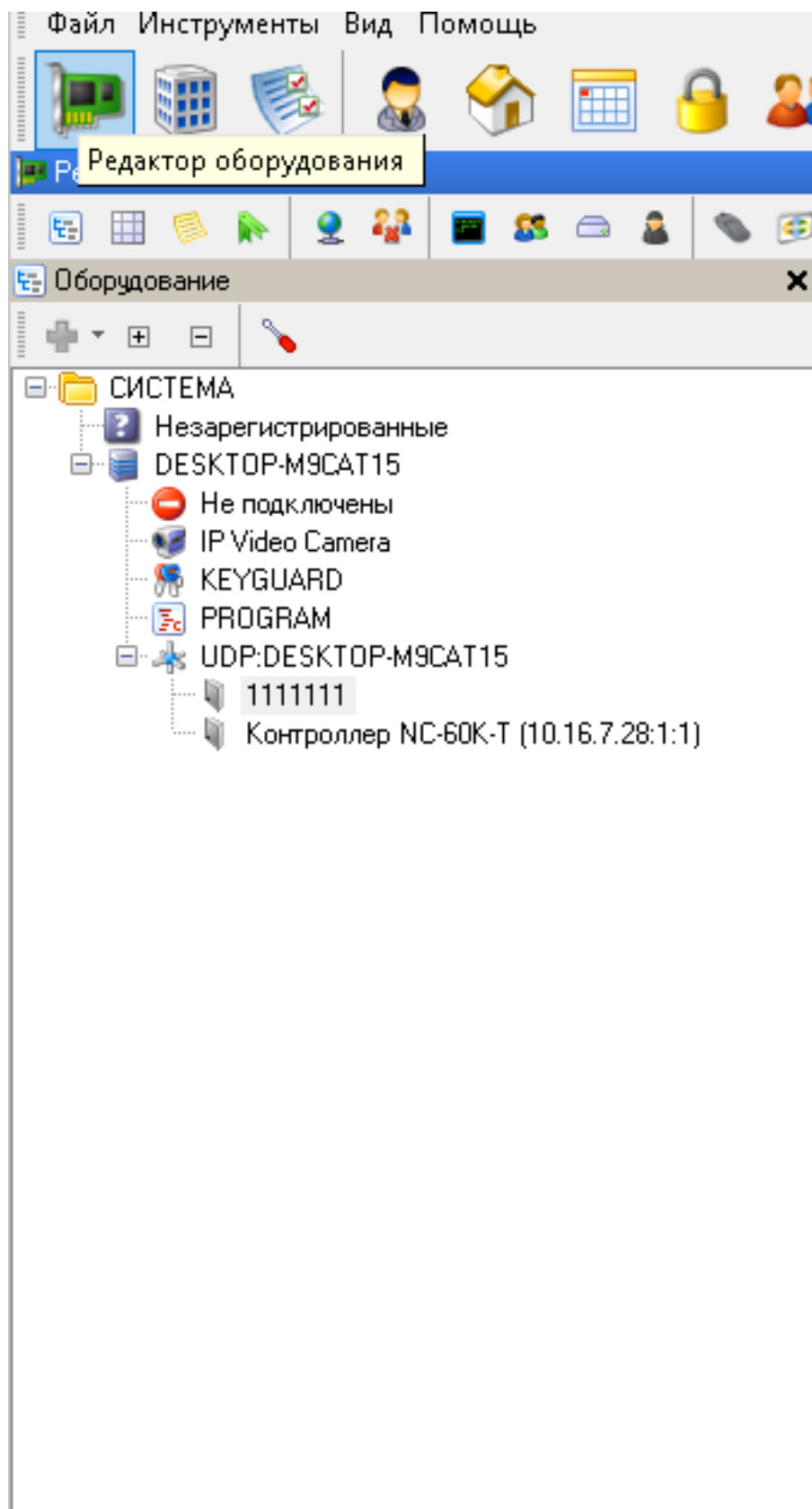


Figure 78. "Equipment editor" section

- For each required controller, set the following settings in the “Access Modes” tab (Figure 79).

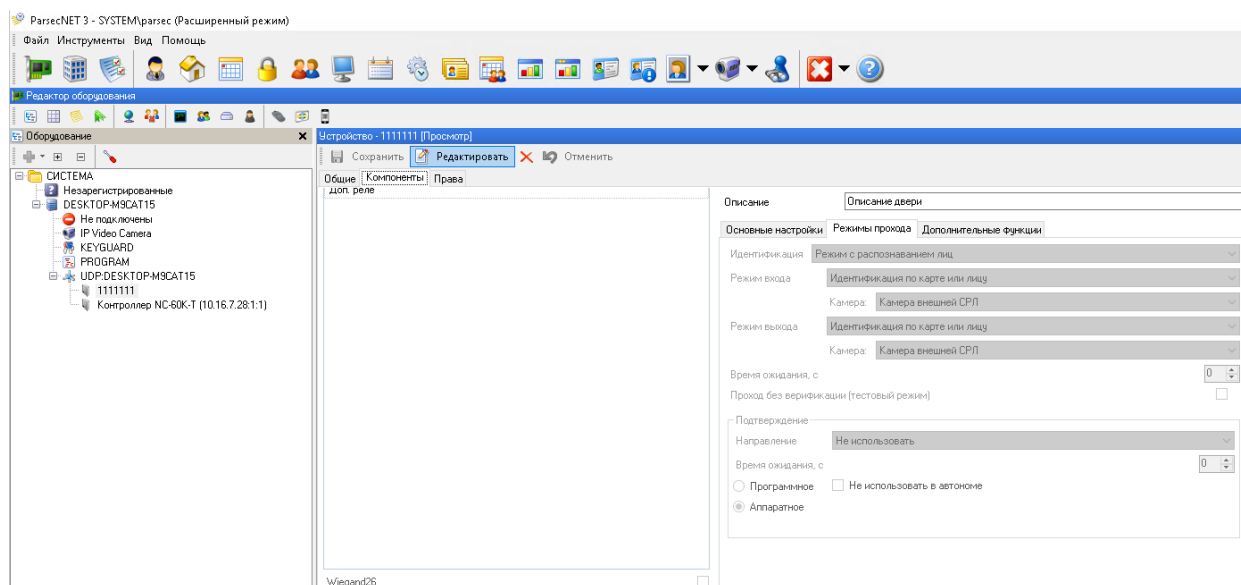


Figure 79. “Access Modes” tab

- Go to the “System settings editor” section, then open the “Face Recognition (ONVIF)” tab (Figure 80).

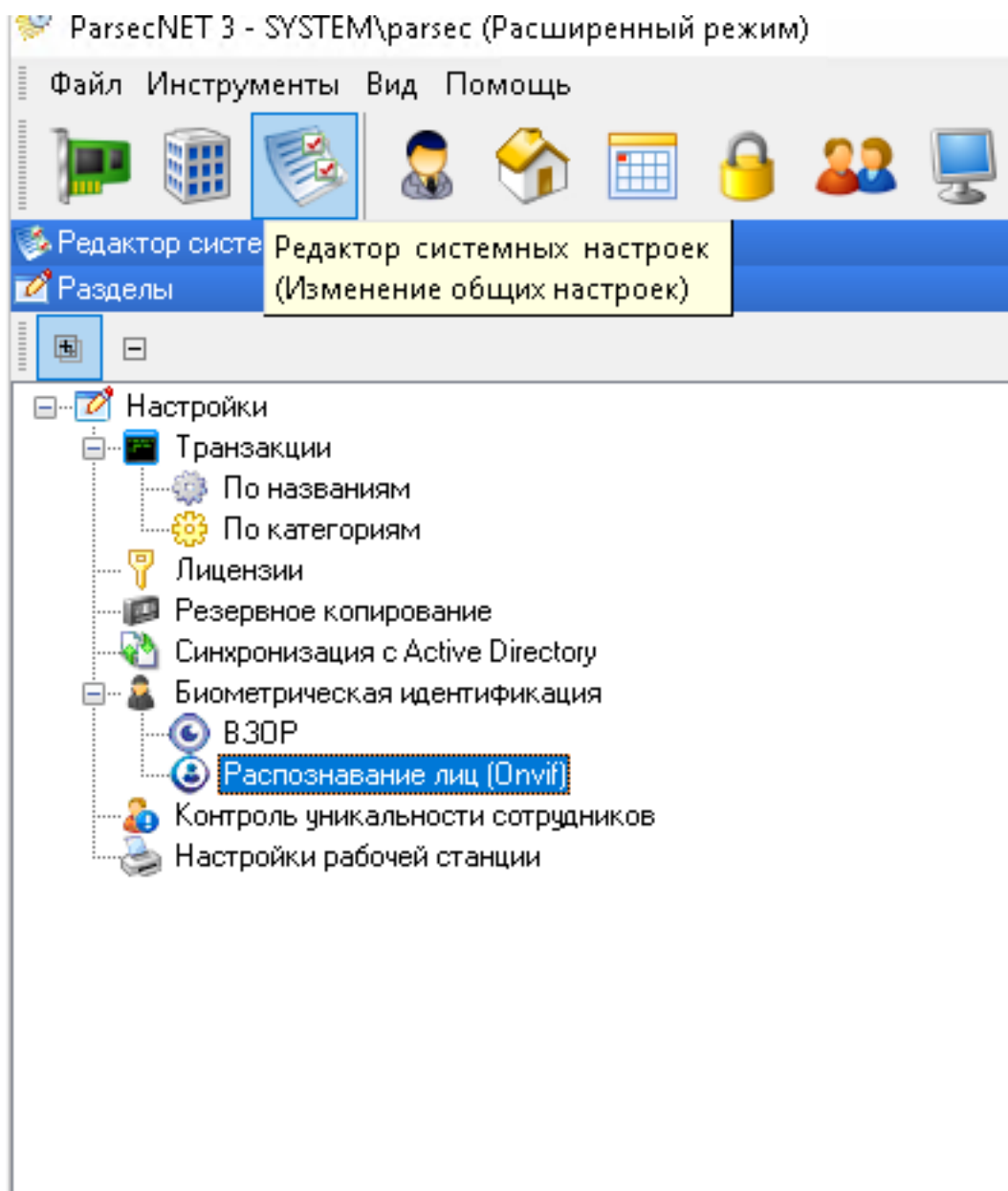


Figure 80. “Face Recognition (ONVIF)” tab

5. In the “Face recognition (ONVIF)” window, click the “Change” button and make sure that the “Use FRS” item is enabled, and the “FRS type” is set to “ONVIF face recognition”.
6. In the “IP Address” and “Port” fields, enter the Access server data.

The IP address of all components must point to the Access server.

7. Click the “Check connection” button only after configuring Access, this will require the “Integration key”.
8. After clicking on the “Check connection” button, the fields in the “Face recognition system services”

block will be filled in automatically.

9. Click the “Save” button.
10. Replicate employees to the Luna list by clicking the “Transfer employees and visitors” button. Before, make sure that all staff members are correctly added in the “Staff Editor” section, see [«Adding staff to Parsec ACS»](#)

Example of displaying staff member unloaded from Parsec ACS to LUNA PLATFORM list (Figure 81).





1 (Количество лиц: 275)				
<input type="checkbox"/>	Информация	Внешний ID	Дата создания	
<input type="checkbox"/>	 <div>Говард Ктулгович Лавкрафт</div>	207d466d7-9782-43c1-84d1-7337778e9a1e	22.09.2023, 14:39:13	  

Figure 81. Displaying staff member in LUNA PLATFORM

13.4. Configuring access groups in Parsec ACS

1. Click the «Access group editor» section.
2. Add a new access group.
3. Add an access territory where the access points are included (Figure 82).

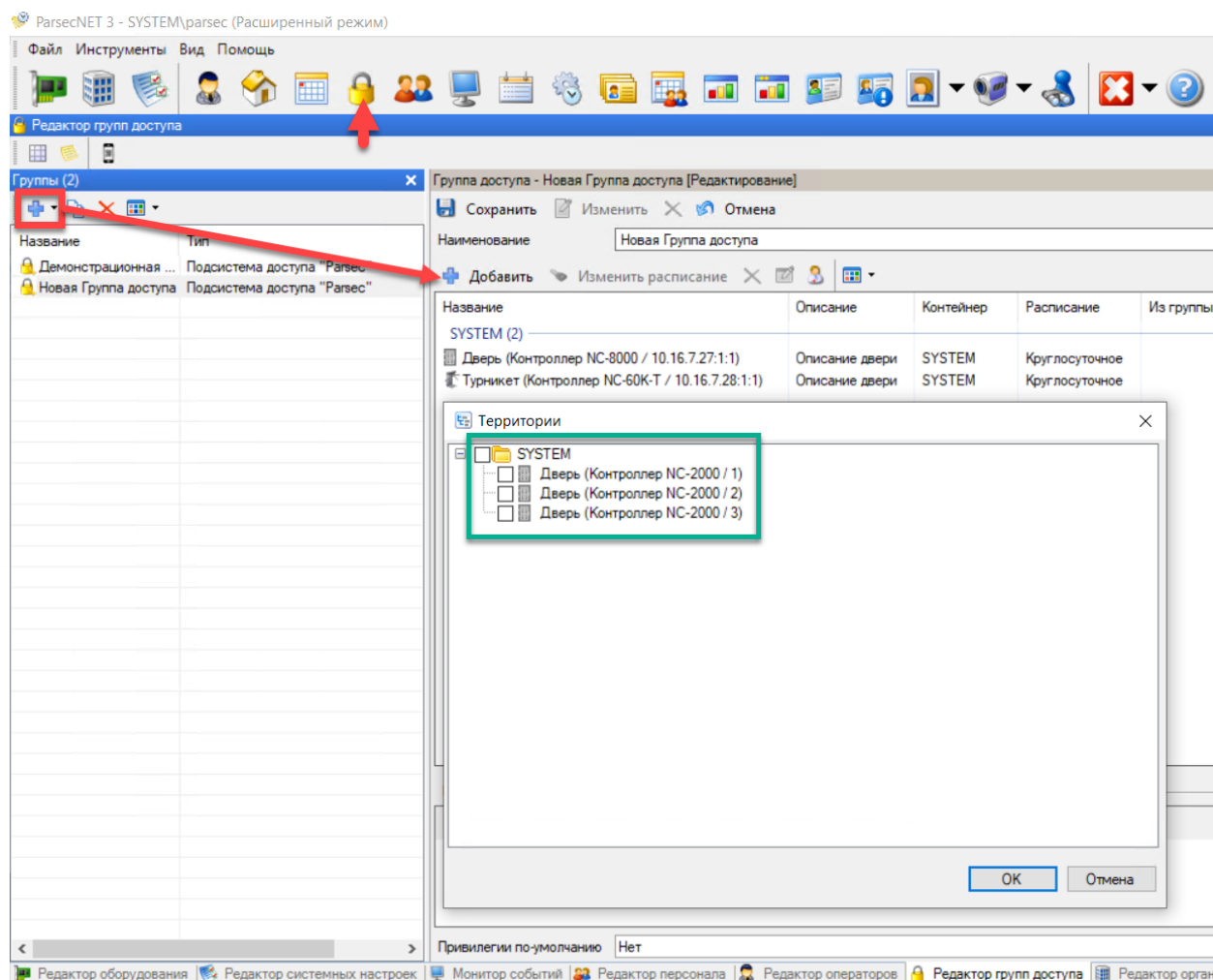


Figure 82. Configuring Access Groups

4. Click the Save button.

13.5. Adding staff to Parsec ACS

Adding staff members to Parsec ACS is necessary for their subsequent upload to LUNA PLATFORM (Figure 83).

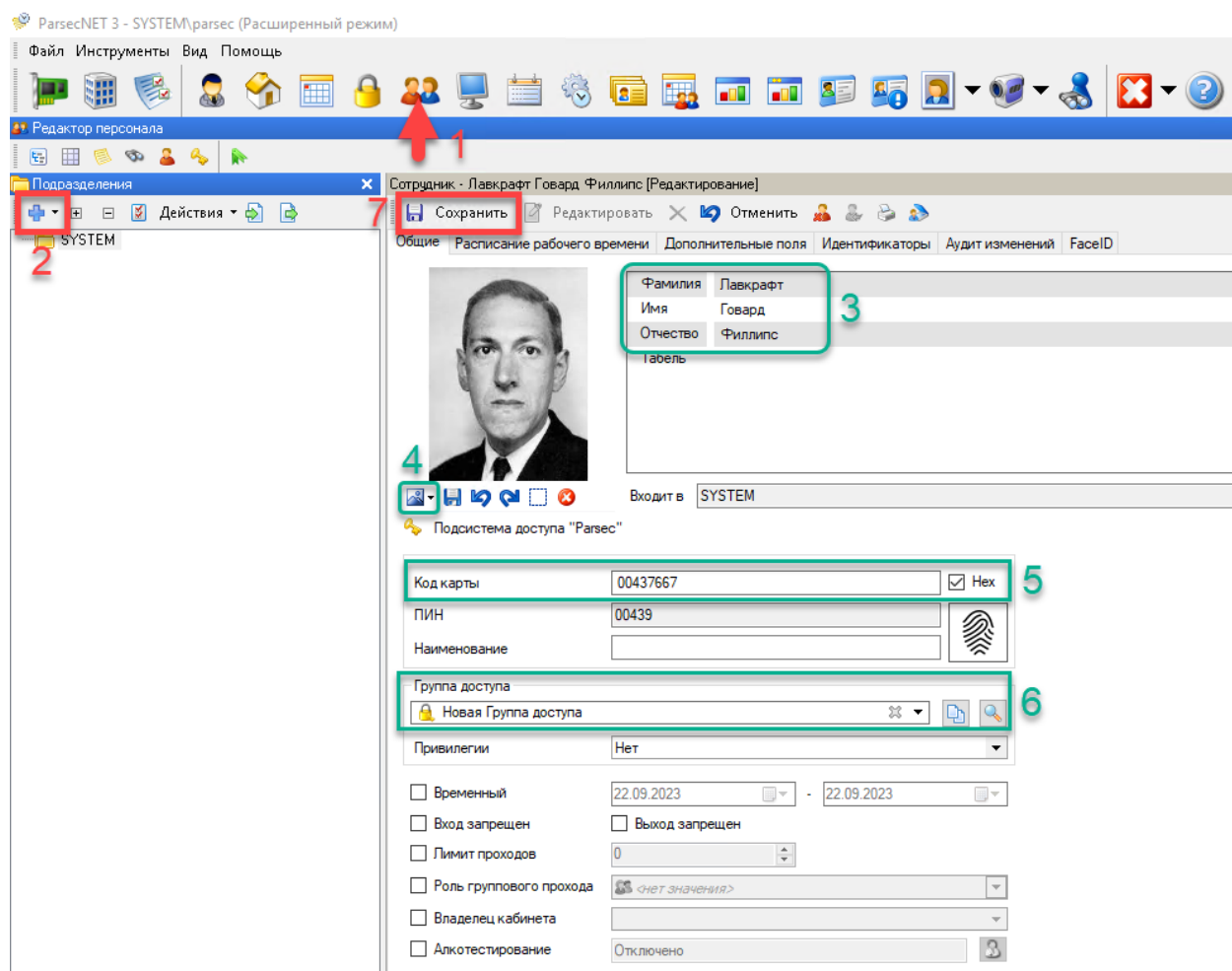


Figure 83. Edit Staff section. Adding a new staff member

1. Click the «Staff Editor» section.
2. Click the button of adding a new staff member.
3. Fill in the «Last Name» and «First Name» fields.
4. Add a photo of the staff member.
5. Fill in the «Card code» field. The «PIN» field will be filled in automatically.

If card access is not provided at the facility or the staff member does not have a card, enter any value in the «Card code» field.

6. Select the staff member's access group.
7. Click the Save button.

If you add staff members correctly, all new or changed data will be added to the LUNA PLATFORM database automatically.

13.6. Methods of interaction with Parsec

Access acts as a server and a client (Table 51).

Sending ONVIF methods to Access occurs at the POST endpoint `/vl-access/webhook/service/onvif/{component_id}`.

Table 51. Parsec methods

Task	Method	Description
Get access points	POST /onvif/accesscontrol	Request to ACS. Getting access point (controller) IDs for manual matching of cameras/terminals and access points
Get a list of ONVIF services	POST /onvif/device_service	Getting a list of component_id ONVIF Access services for connection
Create user	CreateCredential	ONVIF method
Update user	ModifyCredential	ONVIF method
Delete user	DeleteCredential	ONVIF method
Create subscription	CreatePullPoint Subscription	ONVIF method. Subscribe to events.
Get detection events	PullMessages	Receiving an employee detection event. The request is sent every 10 seconds and waits 10 seconds until a frame appears.

13.7. Parsec interaction process diagrams

13.7.1. Connecting the Parsec service

Sequence diagram (Figure 84).

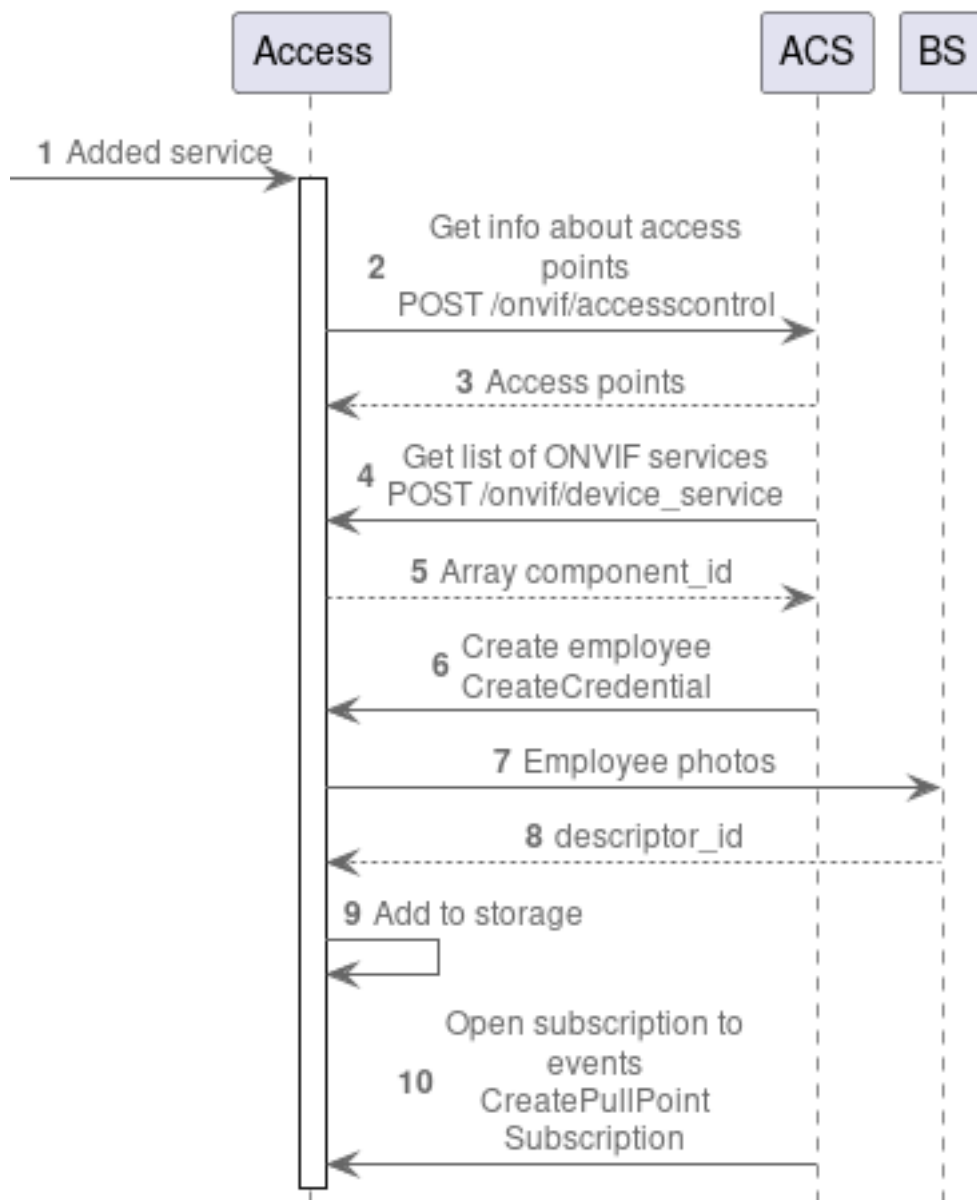


Figure 84. Process diagram for connecting the ACS

1. The user added the Parsec service to Access.
2. Access sends a request to the ACS to obtain access points. The obtained access points are displayed in the info field of the service properties. The request is used to check the availability of the ACS.
3. The ACS returns the access points.
4. The ACS sends a request to Access to obtain a list of Access services that support the ONVIF protocol.
5. Access returns the component_id of the ONVIF services.
6. The ACS sends a POST /vl-access/webhook/service/onvif/{component_id} CreateCredential

request to Access to transfer employees to the Access repository.

7. Access sends a request with employee photos to the BS to retrieve descriptor_id (face_id).
8. The BS returns descriptor_id.
9. Access saves information on each employee to local storage.
10. The ACS sends a request to Access to open a subscription for receiving events (best frames of a person at the terminal).

13.7.2. Parsec event processing with 2 factors

Sequence diagram (Figure 85).

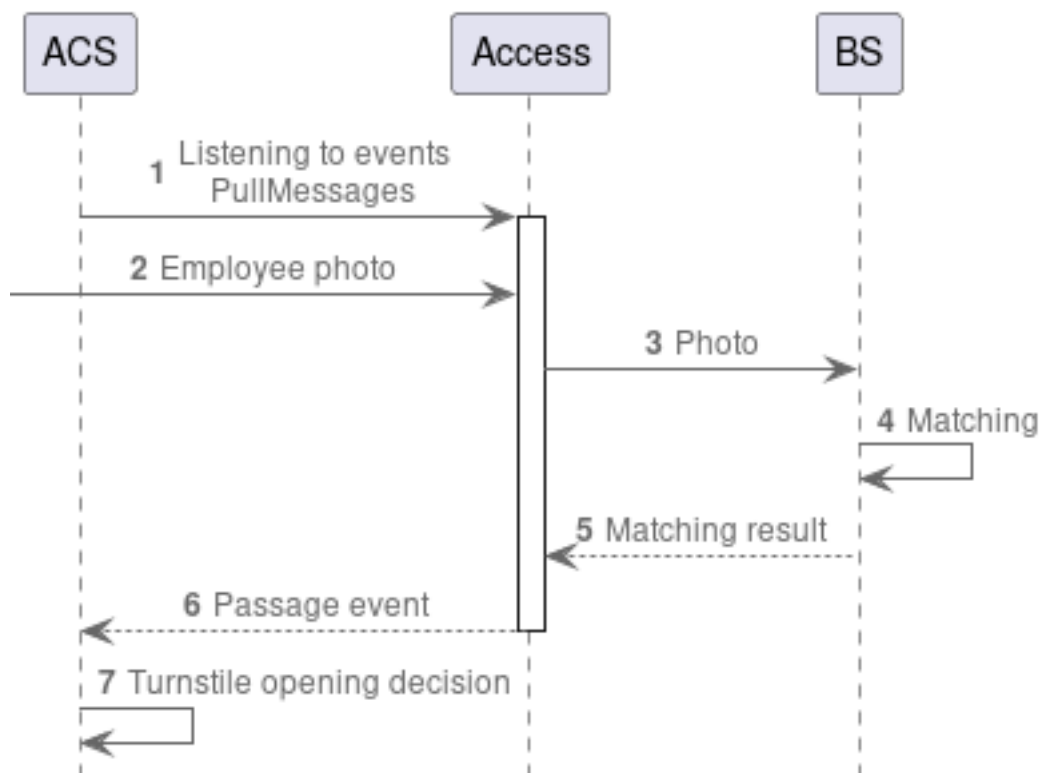


Figure 85. Process diagram with 2 factors

1. ACS sends a POST /vl-access/webhook/service/onvif/{component_id} PullMessages request every 10 seconds to wait for a passage event.
2. Access receives the best photo of the employee at the terminal.
3. Access sends a photo of the employee to the Biometric System.
4. BS compares the photos from the terminal and the one saved in the database.

5. BS returns to Access a decision on granting access.
6. Access returns a passage event to ACS.
7. ACS makes a decision on opening the terminal.

14. PERCo-Web ACS

Software integration of the PERCo-Web ACS software with LP5 is implemented to ensure the passage of recognized persons through a turnstile/door with a magnetic lock.

Supports PERCo-Web system version 2.0, build number 4.30.

Performs user data replication from the PERCo ACS to the specified Luna list and generates PercoController controllers from the received list of devices for execution of entry or exit requests.

14.1. Supported integration options for PERCo-Web ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

Transfer of user data from the ACS to LP5 occurs using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

For the synchronization/replication settings, see the service settings.

Each integration with LP5 (Table 52) uses the [Luna](#) service.

If the terminal does not have data output facilities (e.g., a screen), the [SendToDevice](#) pipeline is not required.

Table 52. LP5 integration options

Service	Device	Pipeline
PercoWeb + PercoController	Beward	MatchByPhoto + SendToController + SendToDevice
	BioSmart	MatchByPhoto + SendToController + SendToDevice
	Dahua	MatchByPhoto + SendToController
	HikvisionCamera	MatchByPhoto + SendToController
	LunaFast4A1	MatchByPhoto + SendToController + SendToDevice
	UniUbi	MatchByPhoto + SendToController + SendToDevice
	VKVision02	MatchByPhoto + SendToController + SendToDevice
	R20Face	MatchByPhoto + SendCardToR20Face / SendToController + SendToDevice

14.2. Standard integration using PERCo-Web

Only 1-factor integration is supported (Figure 86) and (Table 53).

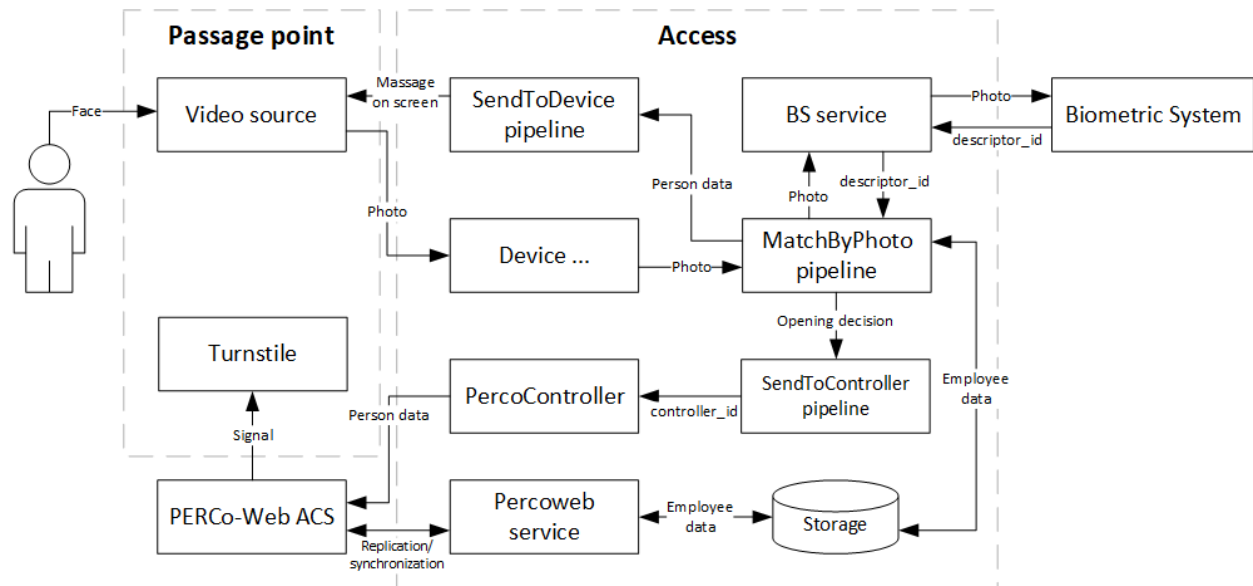


Figure 86. Component diagram for 1-factor integration

Table 53. Integration description

Component	Description
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Video source	A device for extracting a frame of a person's face. Can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream. A biometric terminal allows you to create feedback to show a person information about the passage.
Device ...	An Access component for receiving data from a video data source. Selected based on the device used.
MatchByPhoto pipeline	Access component for interacting with the BS. When working with a biometric terminal, it is necessary to additionally connect the pipeline SendToDevice
BS Service	Access component for interaction with the BS: for LP5 it is Luna , for KBS - the corresponding KBS service.

Component	Description
Biometric system	A system for comparing a reference photo of a person with the best frame obtained from a video data source. Can be either LP5 or supported by CBS.
Storage	A database within Access for storing information about employees.
SendToController Pipeline	Access Component passes the employee ID to PercoController after matching the person and confirming the card number in Access.
PercoController	Access component for sending a card number to the ACS.
PercoWEB ACS	Central software for working with PercoWEB. Stores employee data and makes decisions about granting access.
Turnstile	Barrier device for access control
PercoWEB service	Access component for replicating/synchronizing employees from the ACS and listening to ACS events.

14.3. Methods of interaction with PERCo-Web

Start of endpoint for all requests (Table 54): /api.

Table 54. PERCo-Web methods

Task	Method	Description
Log in	POST /system/auth	Access authorization in ACS. Authorization occurs when adding a service to obtain a token. Token lifetime is 840 seconds.
Availability check	GET /system/language/	ACS availability check. Runs once per minute
Get controllers	GET /devices	Get device_id of controllers to create in Access PercoController.
Get information about the controller	GET /devices/{device_id}	Get information about the controller by its id, if it is active.
Employee synchronization	GET /users/staff/table	Get information about employees: photo availability, activity status, full name and person_id.
Get employee photos	GET /users/{user_id}/image	Get an employee photo

Task	Method	Description
Get events	GET /eventsystem	Request to get employee change events. The request is sent every 10 seconds.
Open the turnstile	POST /devices/{device_id}/pass	Sending a request to open access to a person on the same controller from which the event came.

14.4. Diagrams of interaction processes with PERCo-Web

14.4.1. Connecting the PERCo-Web service

Sequence diagram (Figure 87).

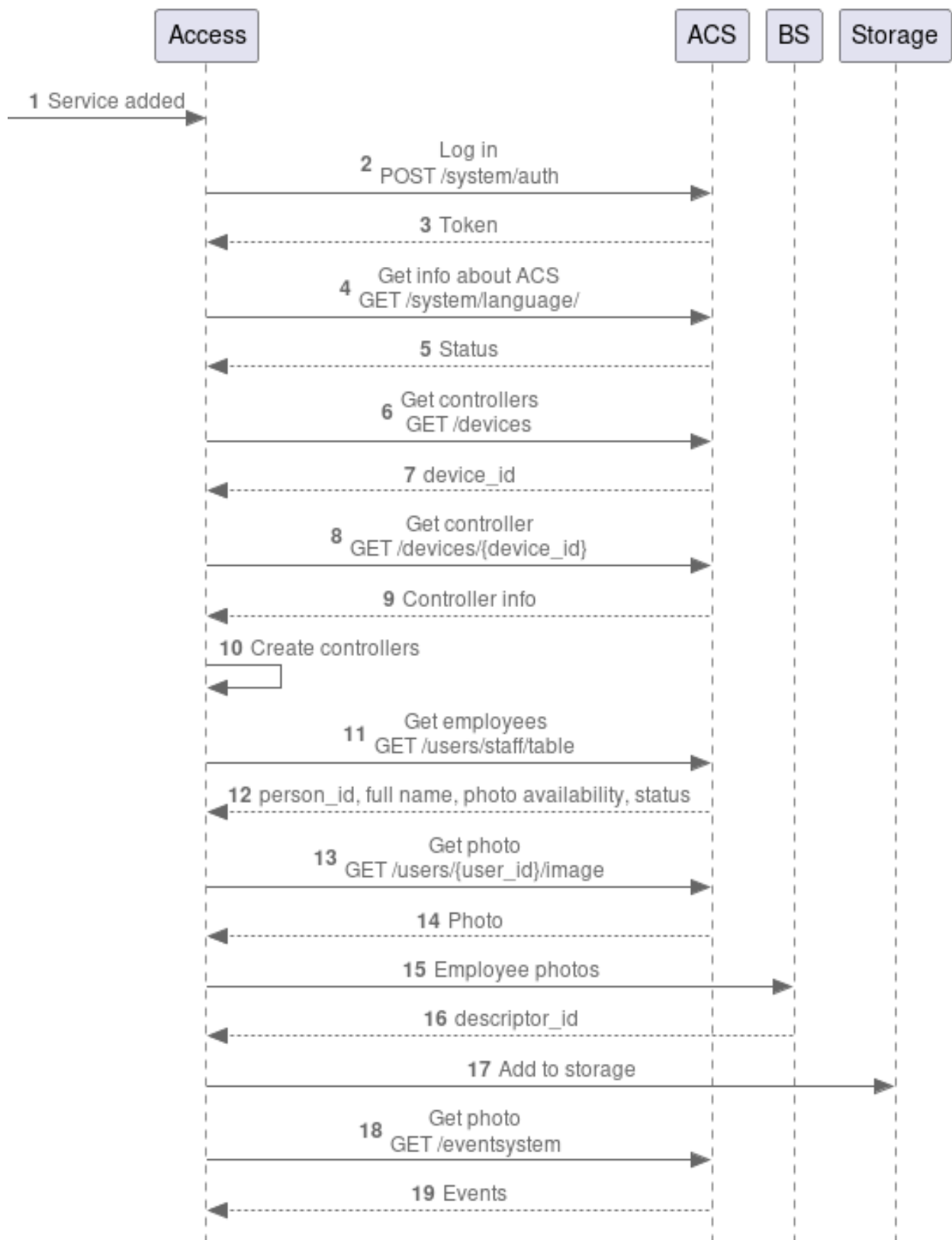


Figure 87. Diagram of processes when connecting ACS

1. The user added the PERCo-Web service to Access.
2. Access sends a request for authorization to the ACS.
3. The ACS returns a token for authorization. The token has a lifetime, after which Access re-performs authorization.
4. Access sends a GET /system/language/ request to determine whether the service is active.
5. The ACS returns a response.
6. Access sends a request to obtain a list of active controllers.
7. The ACS returns an array of device_id.
8. Access sends a request to obtain information about the controller, for each received device_id.
9. ACS returns controller data.
10. Access creates PercoController based on the number of received device_ids.
11. Access sends a request to replicate employees from ACS.
12. ACS returns employee data.
13. Access sends a request to receive photos of employees who are active and have photos.
14. ACS returns employee photos.
15. Access sends a request with employee photos to the BS to retrieve descriptor_id (face_id).
16. BS returns descriptor_id.
17. Access saves employee data in storage.
18. Access sends a request every 10 seconds to receive events about employee changes to perform synchronization.
19. ACS returns events.

14.4.2. PERCo-Web event processing with 1 factor

Sequence diagram (Figure 88).

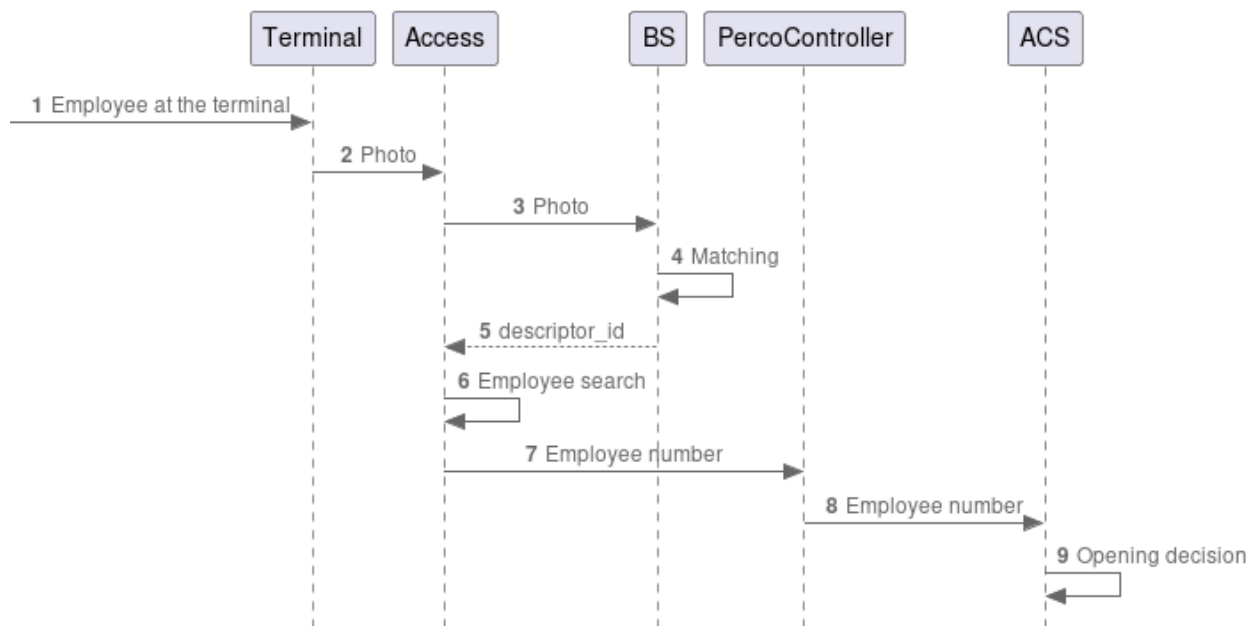


Figure 88. Process diagram with 1 factor

1. Employee at the biometric terminal at the checkpoint.
2. The terminal sends the best shot of the employee to Access.
3. Access sends the employee's photo to the Biometric System.
4. The BS compares the photo from the terminal and the one saved in the database.
5. The BS returns the matching result to Access.
6. Access searches for the employee's number based on the received descriptor_id.
7. Access sends the employee's number to the ACS.
8. The ACS makes a decision to allow the person through.

15. Rusguard ACS

Software integrations of Rusguard ACS with biometric systems are implemented to ensure the passage of recognized persons through the turnstile.

The service replicates employees from the ACS to its database by requesting the descriptor ID in the CBS based on the employee's photo. To update the data, the replication session is restarted 5 seconds after completion.

- Supports versions: System - 3.3.1, Database - 3.3.1

15.1. Supported integration options for Rusguard ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

Transfer of user data from ACS to LP5 occurs using replication - the mechanism for the initial transfer of user data.

For the replication settings, see the service settings.

Each integration with LP5 (Table 55) uses the [Luna](#) service.

If the terminal does not have data output facilities (e.g., a screen), the [SendToDevice](#) pipeline is not required.

Table 55. LP5 integration options

Service	Device	Pipeline
Rusguard + GateController / PusrController	Beward	MatchByPhoto + SendToController + SendToDevice
	BioSmart	MatchByPhoto + SendToController + SendToDevice
	Dahua	MatchByPhoto + SendToController
	Dahua Thermo	MatchByPhoto + SendToController
	Fortuna315	MatchByPhoto + SendToController
	HikvisionCamera	MatchByPhoto + SendToController
	HikvisionCamera Thermo	MatchByPhoto + SendToController
	HikvisionTerminal Thermo	MatchByPhoto + SendToController + SendToDevice

Service	Device	Pipeline
	LunaFast4A1	MatchByPhoto + SendToController
	Panda	MatchByPhoto + SendToController
	UniUbi	MatchByPhoto + SendToController + SendToDevice
	VKVision02	MatchByPhoto + SendToController + SendToDevice
	R20Face	MatchByPhoto + SendToController + SendToDevice

Each integration with CBS (Table 56) uses the CBS service.

Table 56. CBS integration options

Service	Device	Pipeline
CbsMts + Rusguard + LunaStreams	R20Face	MatchByPhoto + SendCardToR20Face + SendToDevice
	BioSmart	MatchByPhoto + GateController / PusrController + SendToDevice
	Dahua	MatchByPhoto + GateController / PusrController
	Dahua Thermo	MatchByPhoto + GateController / PusrController
	Fortuna315	MatchByPhoto + GateController / PusrController
	HikvisionCamera	MatchByPhoto + GateController / PusrController
	HikvisionCamera Thermo	MatchByPhoto + GateController / PusrController
	HikvisionTerminal Thermo	MatchByPhoto + GateController / PusrController + SendToDevice
	LunaFast4A1	MatchByPhoto + GateController / PusrController
	Panda	MatchByPhoto + GateController / PusrController
	UniUbi	MatchByPhoto + GateController / PusrController

Service	Device	Pipeline
	VKVision02	MatchByPhoto + GateController / PusrController + SendToDevice
	R20Face	MatchByPhoto + GateController / PusrController + SendToDevice

15.2. Standard integration using Rusguard

When integrating with Rusguard, standard Access components (Figure 89) and (Table 57) are used.

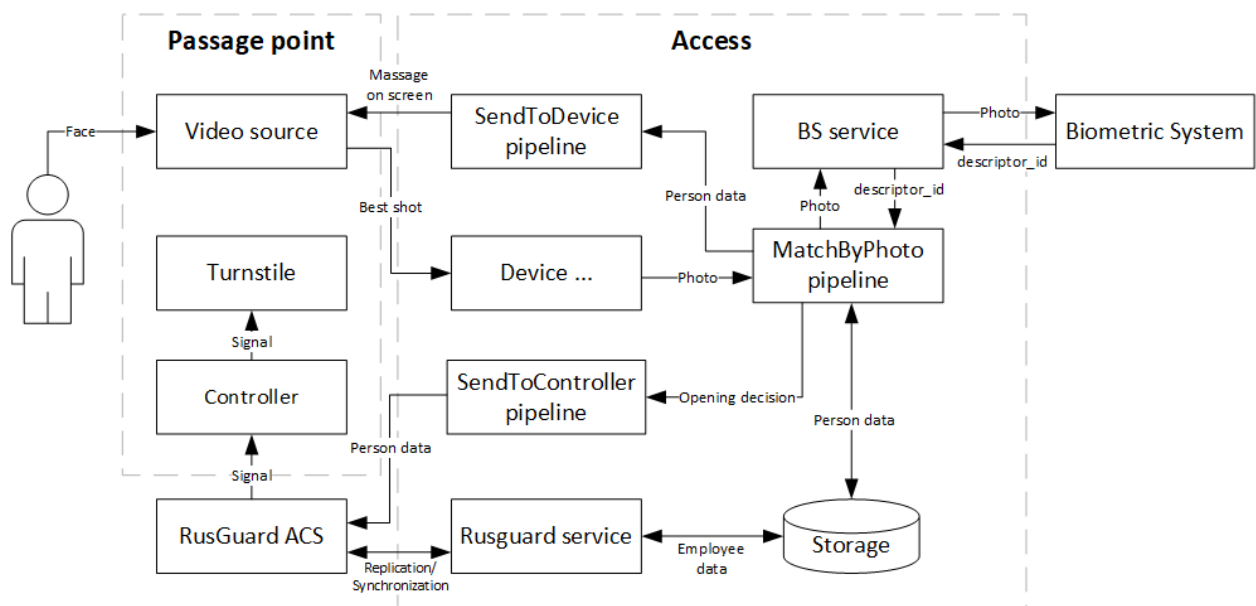


Figure 89. Component diagram for integration with Rusguard

Table 57. Integration description

Component	Description
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access.
Video data source	A device for extracting a frame of a person's face. It can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream (then the LunaStreams service is required).
Turnstile	A barrier device for restricting access.

Component	Description
Rusguard ACS	Central software for working with Rusguard. Stores employee data and makes a decision on granting access.
Rusguard service	An Access component for sending requests and processing data from the ACS.
Device	An Access component for receiving data from a video data source.
Controller	Access point control board.
SendCardToR20Face Pipeline	Access Component for data exchange with ACS
BS Service	Access Component for interaction with BS: for LP5 it is Luna , for CBS - the corresponding CBS service.
MatchByPhoto Pipeline	Access Component for interaction with BS. When working with a biometric terminal, it is necessary to additionally connect the SendToDevice pipeline
Biometric system	A system for comparing a reference photo of a person with the best frame obtained from a video data source. It can be either Luna or a system supported by CBS.
Storage	A database within Access for storing information about employees.

15.3. Methods of interaction with Rusguard

An API is used to exchange data with the ACS (Table 58).

Table 58. Rusguard methods

Task	Method	Description
Get card types	GET /GetCardTypes	Get card types in ACS. Used to check the connection with ACS
Get information about employee photos	GET /GetAcsEmployeePhotoInfos	Get information about employee photos in ACS
Get a list of employee groups	GET /GetAcsEmployeeGroups	Get a list of employee groups in ACS
Get a list of employees from a group	GET /GetAcsEmployeesInGroup	Get a list of employees from a group in ACS

Task	Method	Description
Get an employee photo by ID	GET /GetPhotoEmployee?PersonGuidId= employee_ id&photoNumber=photo_number	Get employee photo in base64 format by his/her ID

15.4. Diagrams of interaction processes with RusGuard

15.4.1. Diagram of interaction between RusGuard ACS and Access

Sequence diagram (Figure 90).

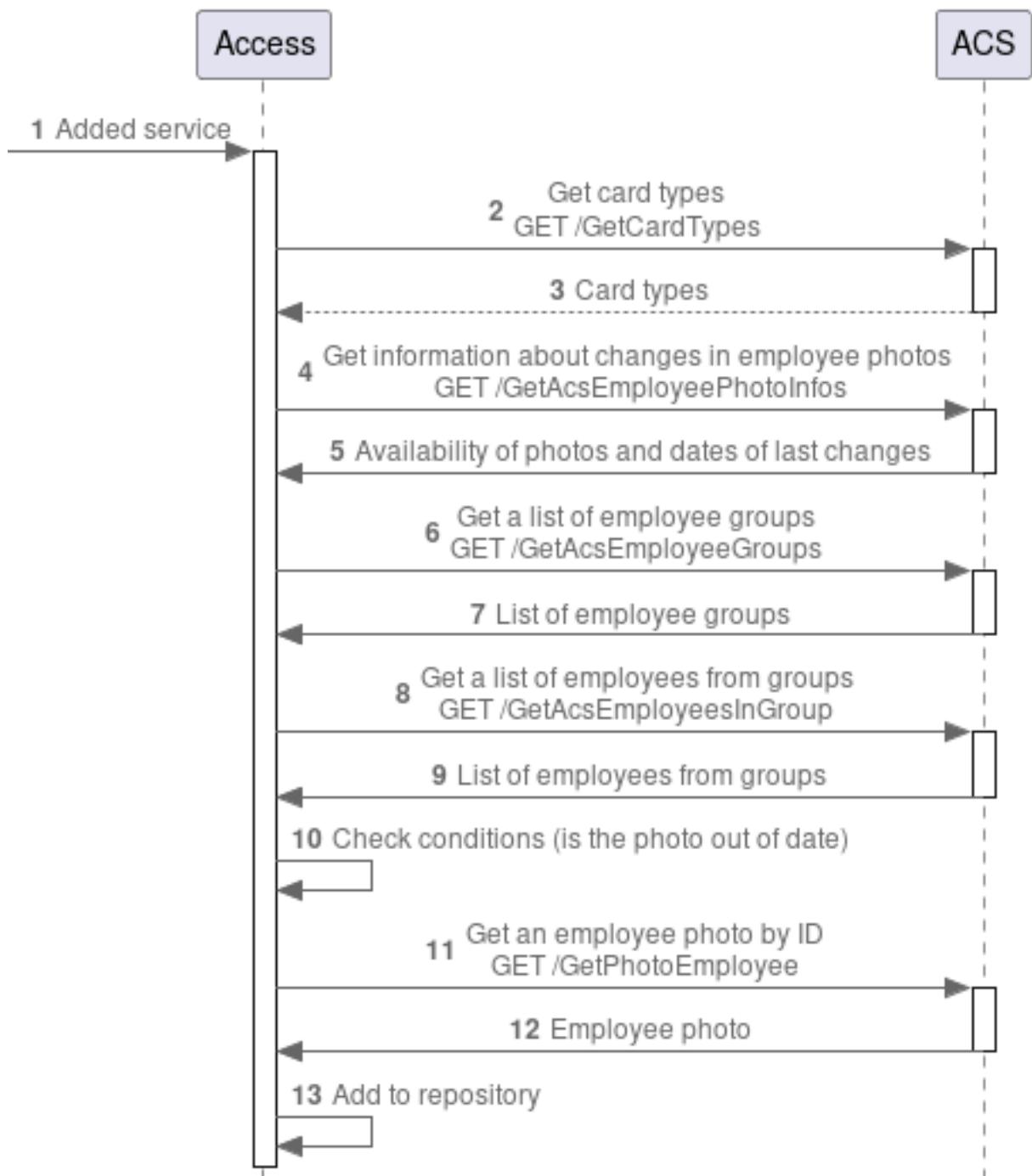


Figure 90. Diagram of interaction between Rusguard ACS and Access

1. The user added the Rusguard service to Access.
2. Access sends a GET /GetCardTypes request to retrieve card types in the ACS, to check the connection.
3. The ACS returns an array of card_types.

4. Access sends a GET /GetAcEmployeePhotoInfos request to obtain information about the availability of employee photos and the dates of their last changes in the ACS.
5. The ACS returns an array of photos.
6. Access sends a GET /GetAcEmployeeGroups request to obtain a list of employee groups in the ACS.
7. The ACS returns an array with group data (ID and group name).
8. Access sends a GET /GetAcEmployeesInGroup request to get employees from a group, for each group from the previous request.
9. ACS returns an array with data for each employee.
10. Access checks if the employee photo is out of date
11. Access sends a GET /GetPhotoEmployee?PersonGuidId=employee_id&photoNumber=photo_number request to get a photo for each employee from the previous request, where:
 - employee_id - employee identifier
 - photo_number - photo number in ACS
12. ACS returns a response with an employee photo in base64 format for each employee.
13. Access saves employee data to the storage.

15.4.2. Diagram of interaction between Access and the biometric system

Sequence diagram (Figure 91).

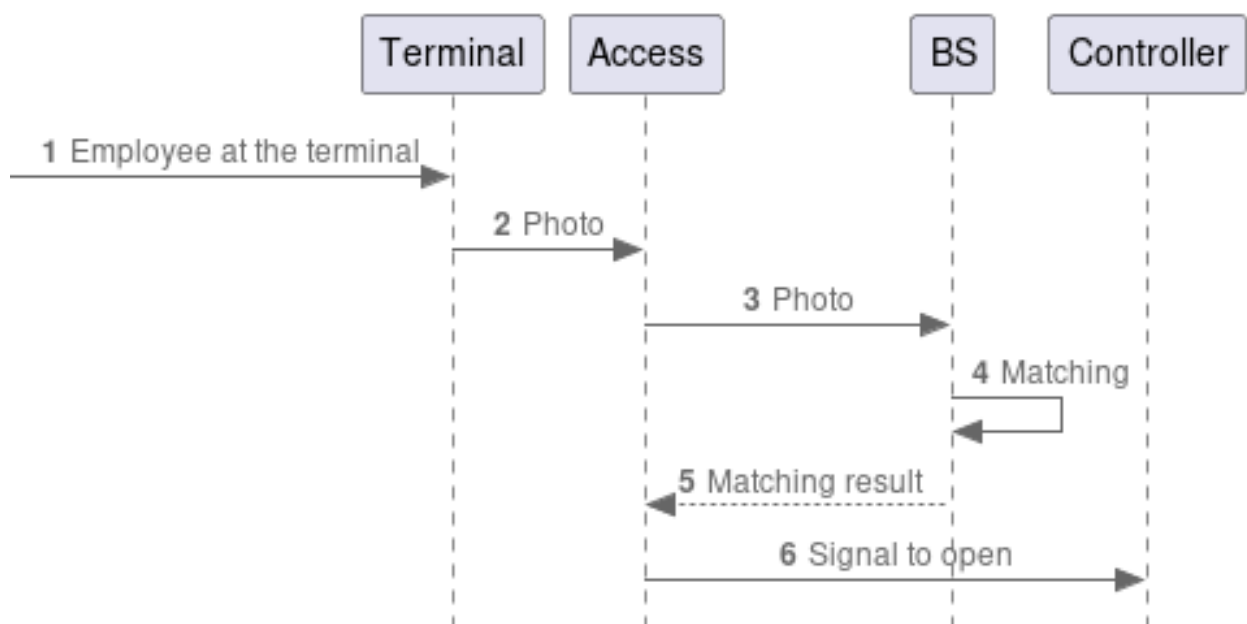


Figure 91. Diagram of interaction between Access and the biometric system

1. An employee at the biometric terminal at the checkpoint.
2. The terminal sends the best shot of the employee to Access.
3. Access sends a photo of the employee to the Biometric System.
4. The BS compares the photo from the terminal and the one stored in the database.
5. The BS returns to Access a decision on granting access.
6. Access determines the employee's access card number from the descriptor received from the BS and sends a signal to the controller to open the access point, specifying exactly this card number. The ACS receives the passage event.

16. SALTO ACS

The ACS synchronizes employees with the list in Luna and listens to events, based on which it decides whether or not to open the turnstile. These events are generated in Access by the SendToSalto pipeline.

- Supports Salto ACS version: 6.6.3.0.

Replicates user data from the Salto database to the specified Luna list and generates SaltoController controllers from the resulting list of access points for subsequent execution of access requests.

For the configuration of the Salto ACS, see the official documentation.

16.1. Supported integration options for SALTO ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

User data is transferred from the ACS to LP5 using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

Each integration with LP5 (Table 59) uses the [Luna](#) service.

Table 59. LP5 integration options

Service	Device	Pipeline
Salto + SaltoController	Beward	MatchByPhoto + SendToSalto
	BioSmart	MatchByPhoto + SendToSalto + SendToDevice
	Dahua	MatchByPhoto + SendToSalto
	HikvisionCamera	MatchByPhoto + SendToSalto
	LunaFast4A1	MatchByPhoto + SendToSalto + SendToDevice
	UniUbi	MatchByPhoto + SendToSalto + SendToDevice
	VKVision02	SendToSalto
	R20Face	MatchByPhoto + SendToSalto + SendToDevice

16.2. Standard integration using Salto

Salto integration (Figure 92) and (Table 60).

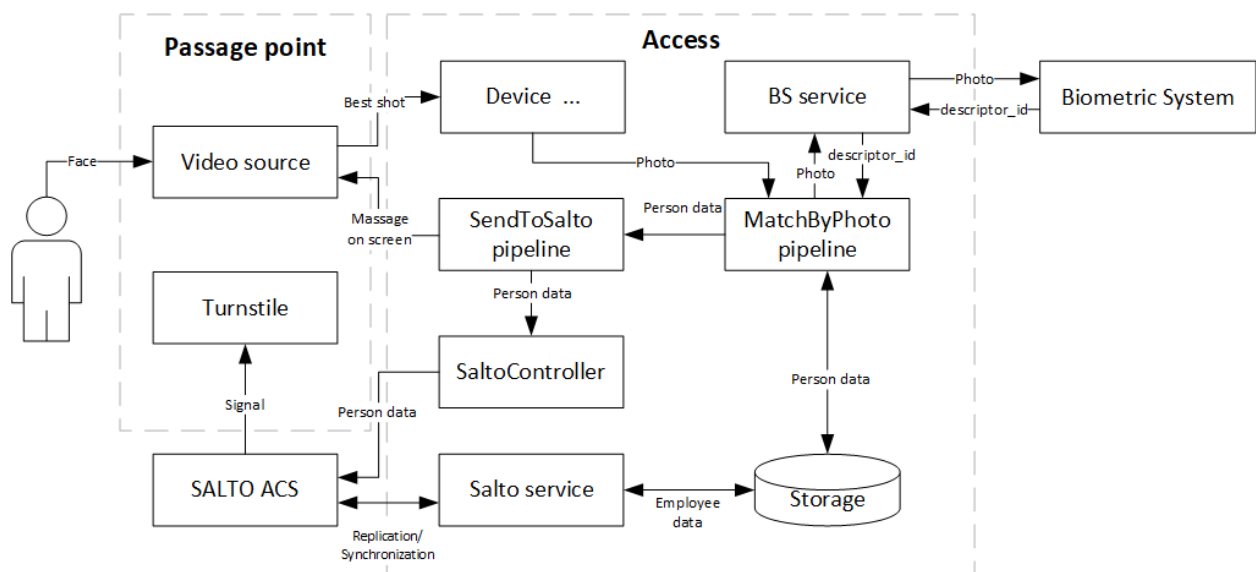


Figure 92. Component diagram for 1f integration

Table 60. Integration description

Component	Description
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Video source	A device for extracting a frame of a person's face. Can be either a biometric terminal (LUNA FAST 4A1 and others). A biometric terminal allows you to create feedback to show a person information about the passage.
Turnstile	A barrier device for access control
SALTO ACS	Central software for working with Salto. Stores employee data and makes a decision on granting access.
Device ...	An Access component for receiving data from a video data source. Selected based on the device used.
SaltoController	Access Component for sending a door unlock request for a specific access point to the ACS.
SendToSalto Pipeline	Access Component for exchanging data with the ACS and transmitting data for display on the device screen.

Component	Description
MatchByPhoto Pipeline	Access Component for interacting with the BS.
Salto Service	Access Component for replicating/synchronizing employees from the ACS and listening to ACS events.
Biometric System	A system for comparing a reference photo of a person and the best frame received from a video data source. Luna is supported.

16.3. SALTO ACS Access Levels

SALTO ACS integrated with LUNA Access has a flexible access level system.

Access levels can be assigned to both doors and employees. Each employee and each door can have a list of access level identifiers.

Information storage:

- Door access levels are stored in the info field of the corresponding controllers in LUNA Access.
- Employee access levels are stored in the local person storage.

Access check:

When attempting to open a door, LUNA Access compares the employee and door access levels. The door will only open if the employee has at least one access level that matches one of the levels assigned to the door.

Integration features:

The request to open the door is sent to SALTO ACS in an anonymous form. The ACS passage events do not display who exactly passed through using biometrics - the event is recorded as the opening of the door by the ACS operator.

16.4. Methods for interacting with Salto

Start of endpoint for all requests (Table 61), except authorization: /rpc

Table 61. Salto methods

Task	Method	Description
Authorization endpoint	/oauth/connect/token	Request for ACS authorization token. Authorization occurs when adding a service
Get ACS version	POST /GetBootstrapConfiguration	Get ACS version

Task	Method	Description
Get ACS access points	POST /GetDoorListStarting FromItem	Request for getting ACS access points
Get a list of employees in ACS (replication)	POST /GetUserListStarting FromItem	Request to obtain a list of all employees in the ACS
Get a list of events in the ACS (synchronization)	POST /GetStatusOfGetSystem AuditorEventList	Request to obtain a list of the latest events in the ACS

16.5. SALTO interaction process diagram

Sequence diagram (Figure 93).

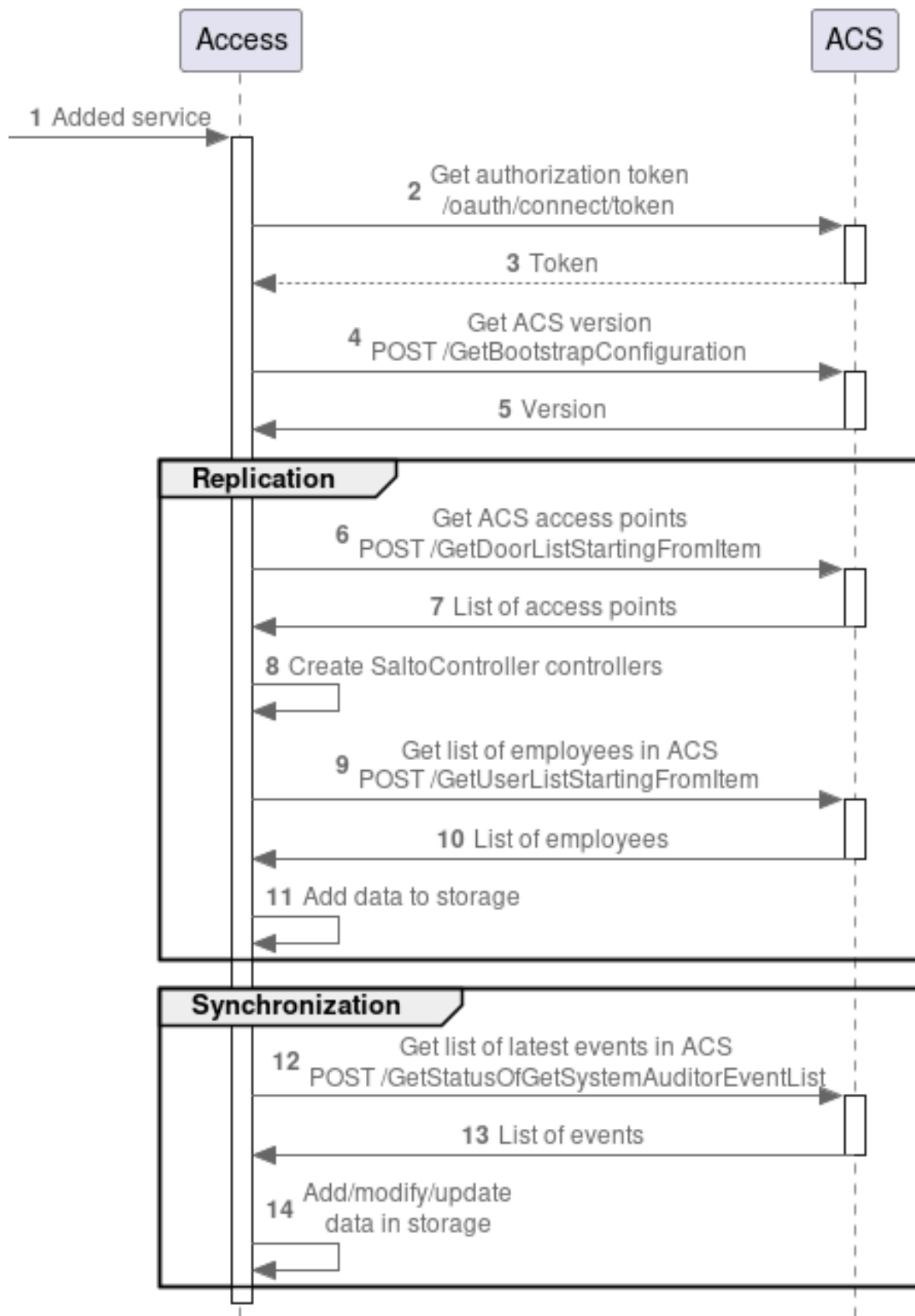


Figure 93. SALTO ACS interaction diagram with Access

1. The user added the Salto service to Access.
2. Access sends a request to obtain an authorization token to the ACS.
3. The ACS returns a token for authorization. The token has a lifetime, after which Access performs authorization again.
4. Access sends a request to obtain information about the ACS.
5. The ACS returns information. Access uses only the ACS version to check compatibility and display the version in the UI.
6. Access sends a request to obtain access points created in the ACS.
7. The ACS returns a list of access points.
8. Access creates SaltoController controllers according to the received IDs.
9. Access sends a request to obtain a list of employees in the ACS.
10. The ACS returns a list of employees.
11. Access saves information on each employee to the local storage.
12. Access sends a request every 10 seconds to receive events about employee changes to perform synchronization.
13. ACS returns events.
14. Access adds/updates information on each employee to the local storage.

17. Sigur ACS

The service is designed to interact with the Sigur ACS. The ACS synchronizes employees with the list in the Biometric System and listens to events, based on which it decides whether to open or not to open the turnstile. These events are generated in the VL Access pipeline `SendToSigur`. The protocol used is HTTP.

- Supports the Sigur ACS version: 1.6.3.18.s.

17.1. Supported integration options for Sigur ACS

Software integrations with Sigur ACS software are implemented for interaction:

- with LP5/CBS for passage of recognized persons through a turnstile/door with a magnetic lock.
- with LUNA CARS for ensuring access control of vehicles when passing through barriers.

The face recognition device generates an event, Access transmits the event to LP5, LP5 processes the event and returns the result to Access for further processing.

User data is transferred from ACS to LP5 using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

For synchronization/replication settings, see the service settings.

If the terminal has no means of outputting data (such as a screen), the `SendToDevice` pipeline is not required.

17.1.1. LP5 Integration Options

Each LP5 integration (Table 62) uses the `Luna` service.

Table 62. LP5 Integration Options

Service	Device	Pipeline
Sigur/ Sigur + LunaStreams	Beward	<code>SendToSigur</code> + <code>MatchByPhoto</code> + <code>SendToDevice</code>
	BioSmart	<code>MatchByPhoto</code> + <code>SendToSigur</code> + <code>SendToDevice</code>
	Dahua	<code>MatchByPhoto</code> + <code>SendToSigur</code>
	Dahua Thermo	<code>MatchByPhoto</code> + <code>SendToSigur</code>
	Fortuna315	<code>MatchByPhoto</code> + <code>SendToSigur</code>

Service	Device	Pipeline
	HikvisionCamera	MatchByPhoto + SendToSigur
	HikvisionCamera Thermo	MatchByPhoto + SendToSigur
	HikvisionTerminal Thermo	MatchByPhoto + SendToSigur + SendToDevice
	LunaFast4A1	MatchByPhoto + SendToSigur
	Panda	MatchByPhoto + SendToSigur
	UniUbi	MatchByPhoto + SendToSigur + SendToDevice
	VKVision02	MatchByPhoto + SendToSigur + SendToDevice
	R20Face	MatchByPhoto + SendToSigur + SendToDevice

17.1.2. Integration options with KBS

Each integration with KBS (Table 63) uses the KBS service.

Table 63. Integration options with KBS

Service	Device	Pipeline
CbsMts + Sigur	Beward	MatchByPhoto + SendToDevice + SendToSigur
	Dahua	MatchByPhoto + SendToSigur
	HikvisionCamera	MatchByPhoto + SendToSigur
	LunaFast4A1	MatchByPhoto + SendToDevice + SendToSigur
CbsAkbars + Sigur	Beward	MatchByPhoto + SendToDevice + SendToSigur
	Dahua	MatchByPhoto + SendToSigur
	HikvisionCamera	MatchByPhoto + SendToSigur
	LunaFast4A1	MatchByPhoto + SendToDevice + SendToSigur
CbsVtb + Sigur	Beward	MatchByPhoto + SendToDevice + SendToSigur
CbsVtb + Sigur + [PersonStorage Actualization]	Dahua	MatchByPhoto + SendToSigur

Service	Device	Pipeline
CbsVtb + Sigur + [CryptoPro]	HikvisionCamera	MatchByPhoto + SendToSigur
	LunaFast4A1	MatchByPhoto + SendToDevice + SendToSigur

Services specified in brackets, for example [CryptoPro], are not mandatory and can be used in integrations if necessary.

17.1.3. Integration options with LUNA CARS

Each integration with the vehicle access control system (Table 64) uses the [LunaCars](#) service.

TC ACS - Access control system for the vehicle territory using a barrier.

LUNA CARS transmits vehicle detection events to Access for further processing.

Cameras are connected via LUNA CARS.

Table 64. Vehicle access control system integration options

Additional service	Pipeline
Sigur	SendCarsToSigur
Sigur + LaurentController	SendCarsToLaurent

17.2. Standard integrations using Sigur

1. Sigur integration diagram for passage of recognized persons through a turnstile/door with a magnetic lock. Standard Access components (Figure 94) and (Table 65) are used for integration with Sigur.

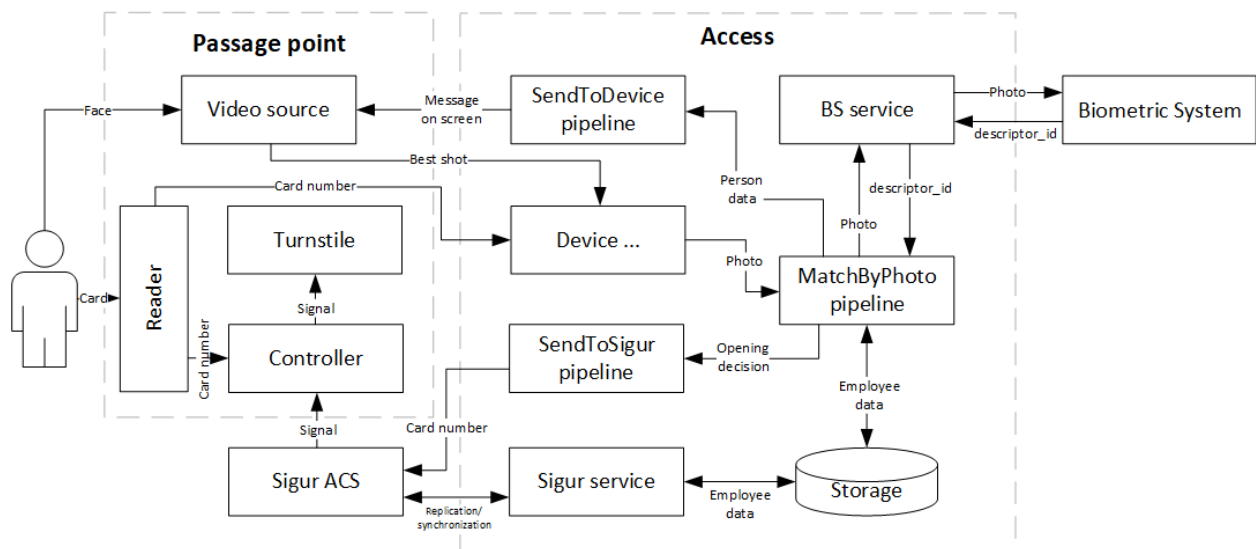


Figure 94. Component diagram for integration with Sigur

Table 65. Integration Description

Component	Description
1φ	
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Reader	A device for receiving access card data.
Controller	A passage point control board.
Turnstile	A barrier device for access control
Sigur ACS	Central software for working with Sigur. Stores employee data and makes a decision on granting access.
Sigur Service	An Access component for processing information from an ACS.
Add-on for 2f	
Video source	A device for extracting a frame of a person's face. It can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream.
Device ...	An Access component for receiving data from a video data source. Selected based on the device used.

Component	Description
Working with LP5 and CBS	
MatchByPhoto pipeline	Access component for interacting with BS
SendToController pipeline	Access component for interacting with CBS
SendToDevice pipeline	Access component for sending a signal to open a relay to a device and displaying text on the screen

2. Sigur integration scheme for providing access control for vehicles when passing through barriers. Standard Access components (Figure 95) and (Table 66) are used for integration with Sigur.

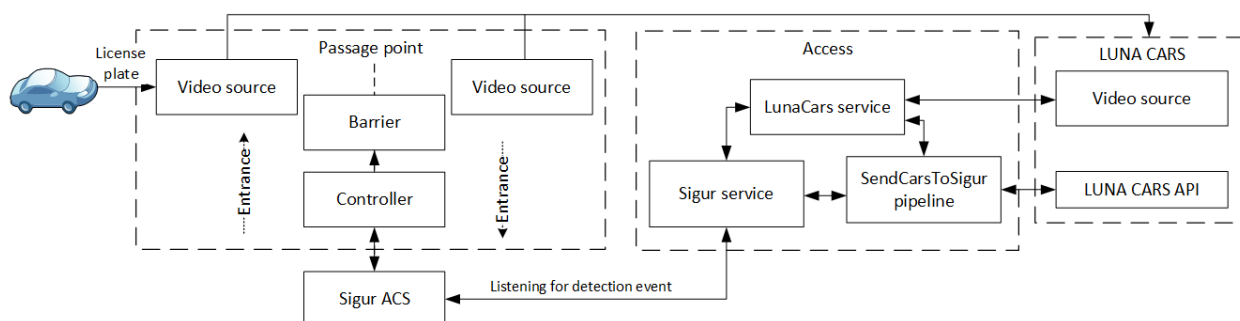


Figure 95. Scheme of components for integration with Sigur

Table 66. Integration description

Component	Description
Vehicle (V)	A car wishing to pass through a point of passage.
Point of passage	A set of components used to control vehicle access. More than one point of passage can be connected, limited by the ACS license. A point of passage can be used for both entry and exit. A separate video data source is used for each direction.
Controller	Passage point control board.
Barrier	A barrier device for access control
Sigur ACS	Central software for working with Sigur. Stores vehicle data and makes a decision on granting access.
Sigur Service	Access component for processing information from ACS.
Video data source	A device for extracting a frame of the vehicle's state registration plate.

Component	Description
Device ...	Access component for receiving data from a video data source. Selected based on the device used.
Working with LUNA CARS	
SendCarsToSigur Pipeline	Access Component for sending events from LUNA CARS to Sigur. Access connects to LUNA CARS Analytics backend using websocket
LunaCars Service	Access Component for hardware and software integration, required for communication between LUNA CARS and blocking devices

17.3. Setting up Sigur ACS software

To launch and set up Sigur ACS software, you must do the following:

- 1. Make sure that you are using Sigur ACS software version 1.6.3.18.s or later:
 - In the Sigur control program menu, select the menu item “Help” → “About the program”.
 - Compare the software version with the one indicated on the website www.sigur.com.
 - If necessary, update the software to the latest version.
- 2. Set up interaction between the integration module and the Sigur ACS software server:
 - In the Sigur control program menu, select the menu item “File” → “Settings”.
 - In the “Edit settings” dialog, go to the “Video surveillance” item (Figure 96).
 - Add a video surveillance server (Figure 97).

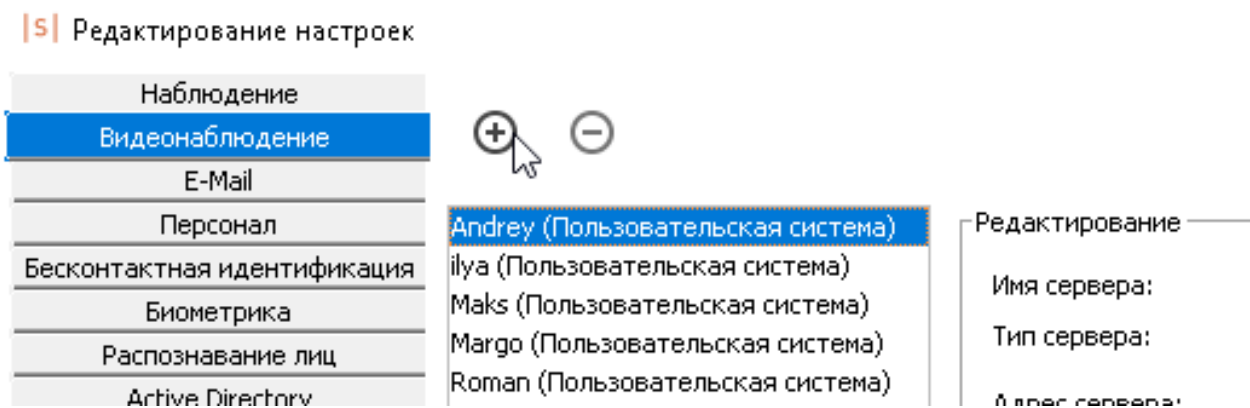


Figure 97. Editing Sigur settings

- Specify an arbitrary server name.
- Server type — «Custom system».

- Click «OK»;

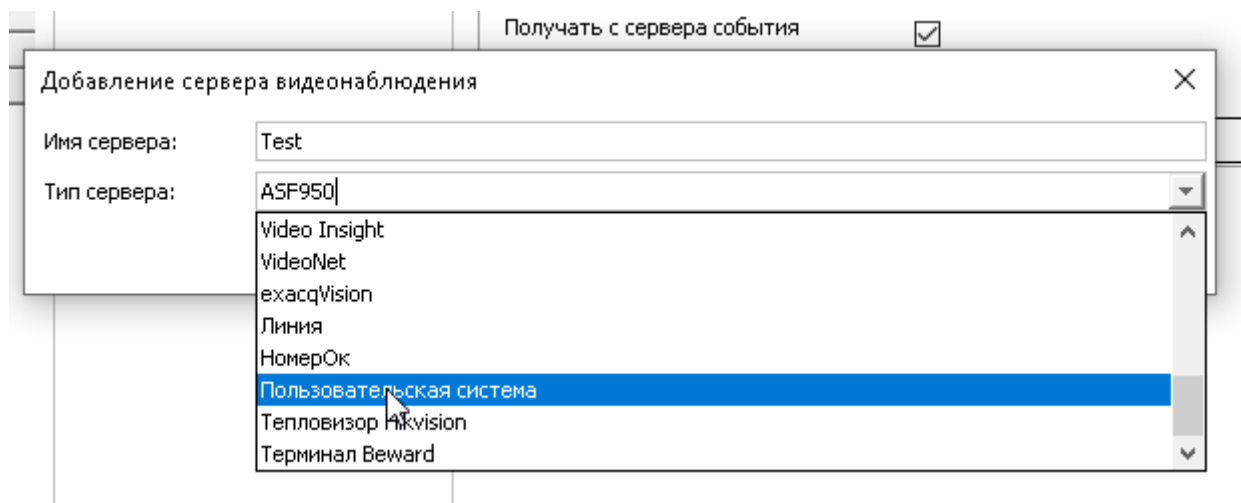


Figure 97. Adding a video surveillance server

3. Configure the server parameters (Figure 98):

Редактирование

Имя сервера:	<input type="text" value="Test"/>
Тип сервера:	<input type="text" value="Пользовательская система"/>
Адрес сервера:	<input type="text" value="10.0.10.161"/>
Порт сервера (HTTP):	<input type="text" value="9091"/>
Путь к сервису:	<input type="text" value="ebhook/service/sigur/f24bc9c5-fa32-4ac0-9728-07bc1178b4d9/"/>
Имя пользователя:	<input type="text"/>
Пароль пользователя:	<input type="password"/>
Аутентификация:	<input type="text" value="отключена"/>
Выгружать на сервер фотографии	<input checked="" type="checkbox"/>
Выгружать на сервер пропуска	<input type="checkbox"/>
Получать с сервера события	<input checked="" type="checkbox"/>
Выгружать на сервер доп. параметры	<input checked="" type="checkbox"/>
Доп. параметры для выгрузки	<input type="text" value="MDM_ID"/> <input type="button" value="Выбрать"/>

Figure 98. Editing video surveillance server parameters

- “Server address” and “Server port (HTTP)” - used when accessing the server from the ACS via HTTP;

- “Server address” corresponds to the IP address of the machine on which Access is running;
- “Server port (HTTP)” - the port for the integration module (the default value is “9091”, if the port is already in use - change);
- “Path to service” specifies the common prefix of paths on the server for all requests from the ACS. This value should be taken from the information block of the Sigur component in Access, the value of the webhook-url field (Figure 99).

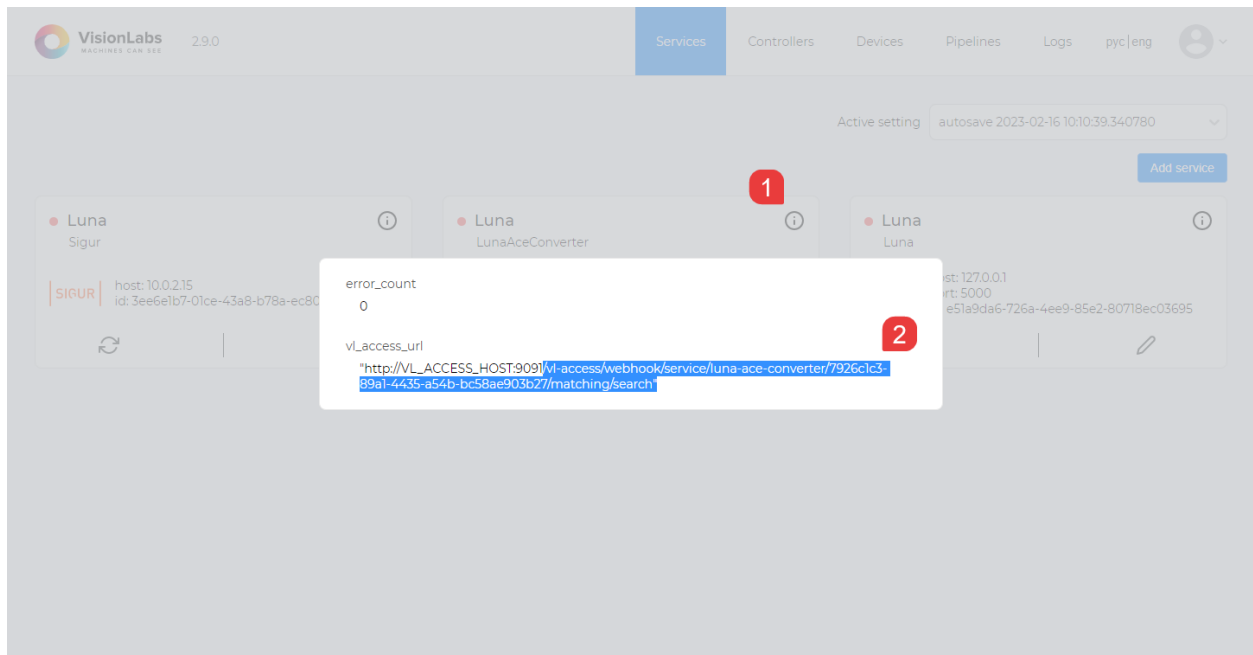


Figure 99. Webhook Link

- Enable the “Upload photos to servers” flag when using the Sigur service, disable when using the SigurThroughDatabase service.
 - Enable the “Receive events from the server” flag.
4. Enable the face recognition function:
- In the “Edit settings” dialog, go to the “Face recognition” item.
 - Check the “Enable face recognition” item (Figure 100).

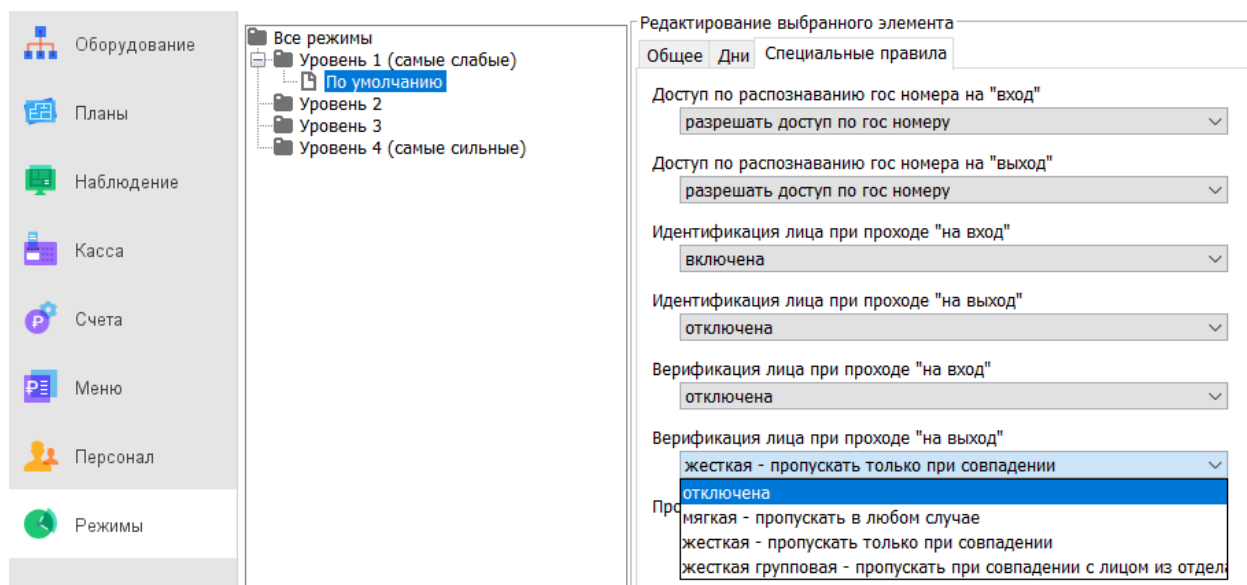


Figure 100. Enabling face recognition in system settings

17.3.1. Setting up access points in Sigur

If errors occur, restart the Sigur ACS software server so that it can connect to the integration module.

To set up access points in Sigur, you must do the following:

1. In the side menu of the Sigur management program, select the “Hardware” item (Figure 101).

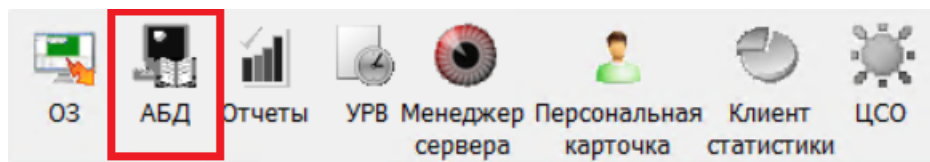


Figure 101. Side menu of the Sigur management program

2. Select the required access point and set up video surveillance parameters for it (Figure 102):

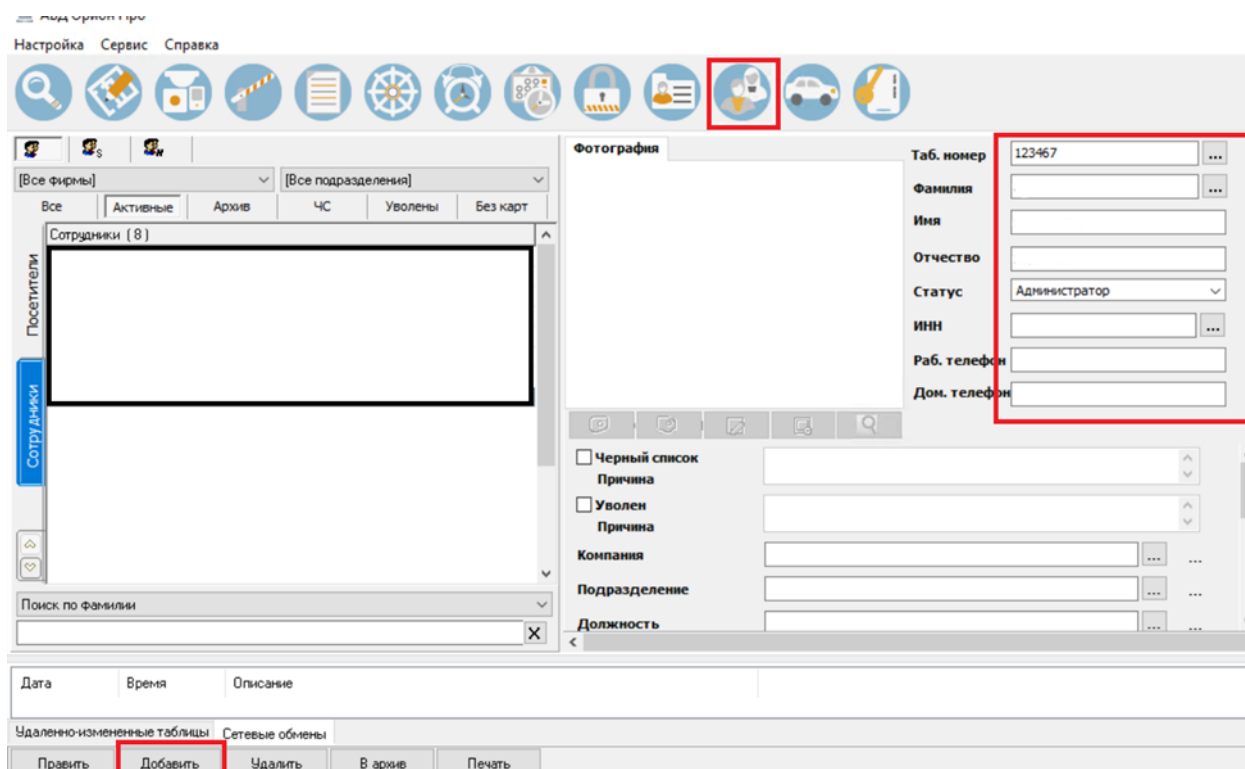


Figure 102. Setting up video surveillance equipment parameters

- “System” - select the name of the created user system;
- “Camera” - select a camera. When you click on the drop-down list, it should display the names of all devices created in Access, this indicates that the integration is working properly and Sigur was able to connect to Access. Select the device that is used to identify the desired access point;
- Activate the flags “Allow face verification” and “Enable face identification”;
- Click the “Apply” button.

After this, replication of employees from Sigur should begin. For a description of the replication algorithm, see the section [Diagram of interaction between Sigur ACS and LUNA Access] (#sigur-interaction).

17.3.2. Setting up access modes in the Sigur ACS software

The ACS has two identification modes in the selected direction:

- 1f mode - based on recognition of the access object’s face
- 2f mode - based on the main identification feature (card) and on the access object’s face

To set up the software, follow these steps:

1. In the Sigur control program, in the “Equipment” side menu, select the required access point and go to the video surveillance settings (Figure 103):

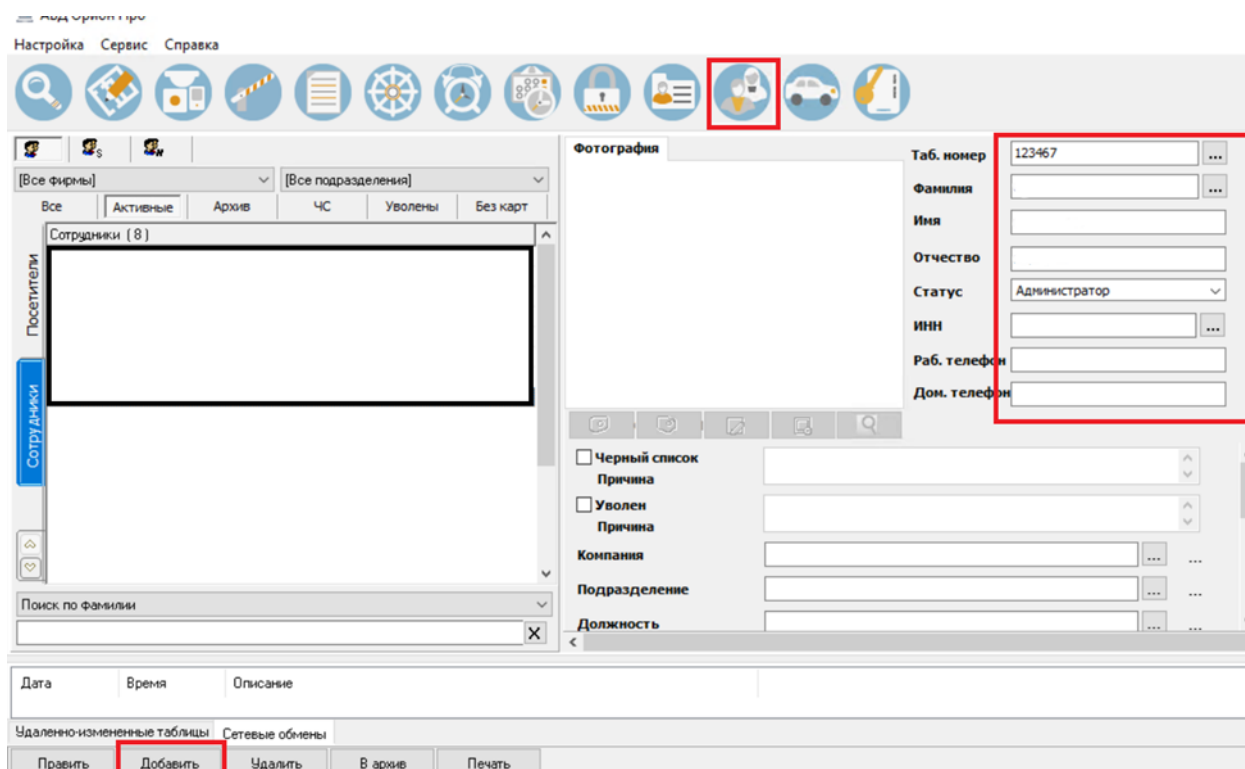


Figure 103. Configuring video surveillance equipment parameters

To enable 1f mode: activate the “Enable face identification” flag.

To enable 2f mode: activate the “Allow face verification” flag.

2. Go to the “Modes” tab and, having created a new one or selected the required one from the existing ones, go to the “Special rules” tab (Figure 104):

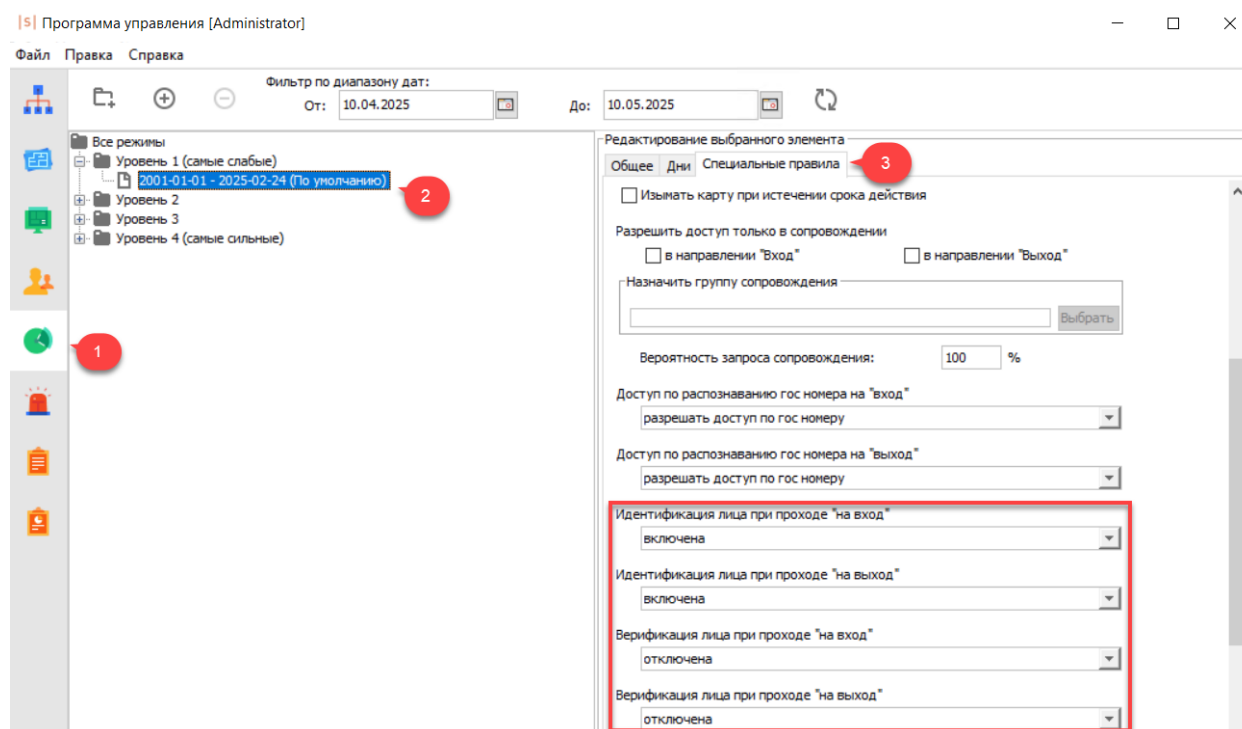


Figure 104. Setting up special rules for passage

For operation in 1-phase mode: If access in any of the directions should be provided only by the fact of face recognition, then for the parameters “Face identification at the entrance/exit” set the value “Enabled”.

For operation in 2-phase mode: If access in any of the directions should be provided by the fact of presenting the employee’s primary ID and the fact of recognition of his face, then for the parameters “Face verification at the entrance/exit” set one of the following values:

- Soft — let in anyway. After identification by the main feature (presenting the card), the system grants access if it is possible according to other criteria, even if the person never appears in the frame. Such access permission will be accompanied by the event “Face not recognized”.
- Hard — let in only if there is a match. After identification by the main feature (presenting the card), the system checks whether the person’s face was recognized in the frame during the time specified in the settings before the event, and if not, waits for the person to appear in the frame within 5 seconds. If the person still does not appear in the frame, access is not granted. The “Surveillance” tab in the software displays the event: “Access denied. Face not identified.”
- Strict group - skip only if matched with a face from the department. After identification by the main feature (presenting the card), the system checks whether the person’s face was recognized in the frame during the time specified in the settings before the event, and if not, waits for the person to appear in the frame within 5 seconds. The detected face is checked for compliance with any of the faces located in the same department as the identified object. If the person still does not appear in

the frame, access is not granted. The “Surveillance” tab in the software displays the event: “Access denied. Face not identified.”

- Soft group — skip if there is no match with the person from the department. After identification by the main attribute, the system grants access if it is possible by other criteria, even if the person of the identified person or any other subject of the same department has not appeared. Such access permission will be accompanied by the event “Person not identified”.

17.4. Methods of interaction with Sigur

An API is used to exchange data with the ACS (Table 67).

Table 67. Sigur methods

Method	Description
GET /event	Receiving detection events. One-way stream connection in which Access broadcasts detection events to the ACS.
GET /getpersons	Get a list of employees from Access. Based on the response of this request, the ACS makes a decision on further adding/updating/deleting employees in Access
POST /updateprson {id, name, photoVersion, photo}	Update employee data in Access
GET /removeperson {id}	Delete employee data in Access
GET /getchannels	Get a list of event sources in Access, used to configure access directions

17.5. Diagram of interaction between Sigur ACS and Access

Access acts as a server, and Sigur acts as a client. After configuring the client, Sigur ACS independently executes all requests to Luna Access (Figure 105).

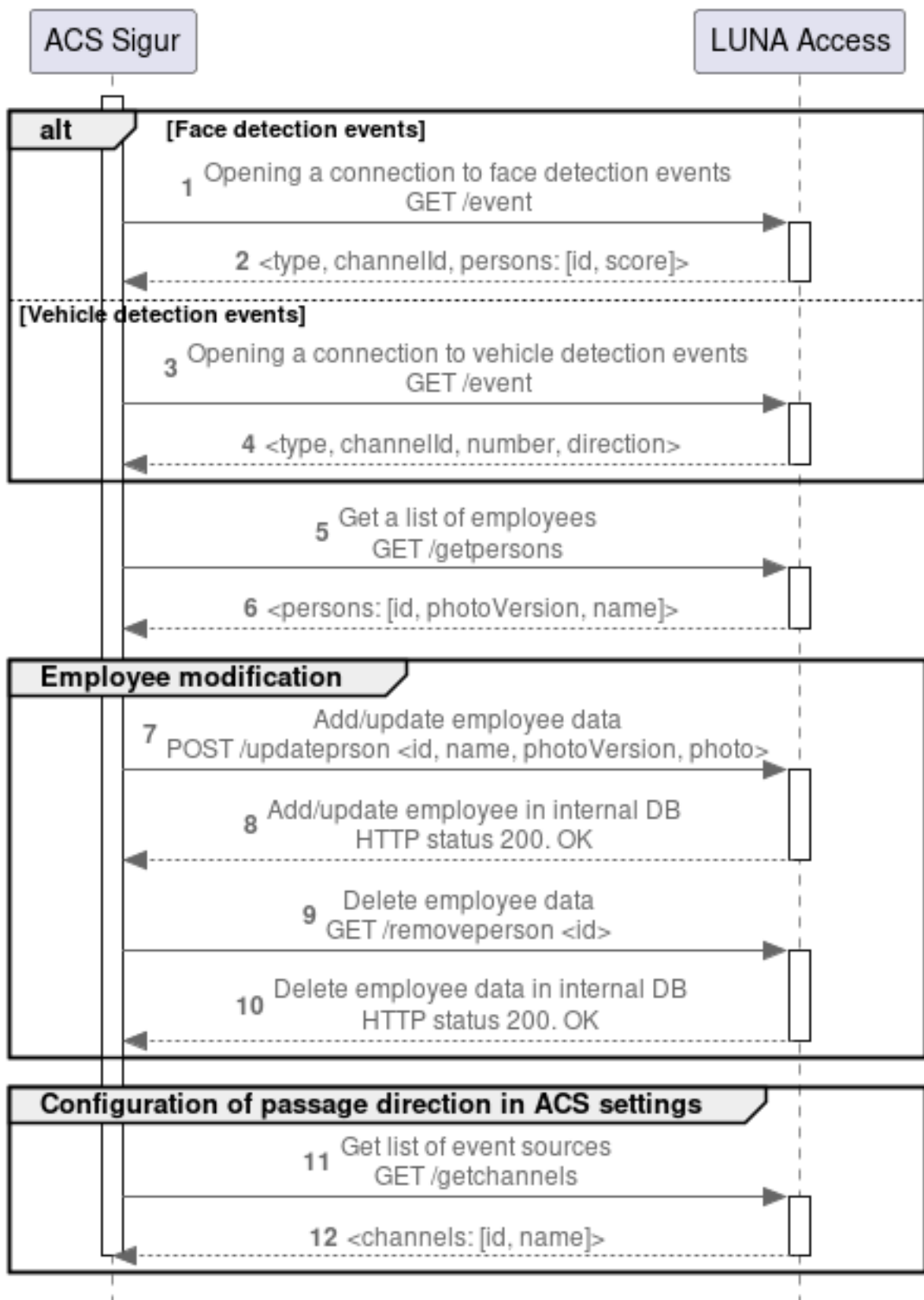


Figure 105. Diagram of interaction between Sigur ACS and Access

Face detection events

1. Sigur ACS initiates a GET request: `/event` to receive face detection events from Access, the connection always remains open.
2. Access returns events to the connection as they occur. Even if there are no events, Access sends an artificial “Keep alive” event every 5 seconds so that both sides of the connection are sure that the connection is still OK. Access returns a json event object with the fields:
 - type - Event type;
 - channelId - Channel ID in Access;
 - persons - Array with recognition results;
 - id - Person ID;
 - score - Similarity level as a real number from 0 to 1.

Vehicle detection events

3. Sigur ACS initiates a GET request: `/event` to receive vehicle detection events from Access, the connection always remains open (Access sends new events to the ACS as they occur).
4. Access returns events to the connection as they occur. Even if there are no events, Access sends an artificial “Keep alive” event every 5 seconds so that both sides of the connection are sure that the connection is still OK. Access returns a json event object with the fields:
 - type - Event type;
 - channelId - Channel ID in Access;
 - number - String of the recognized vehicle registration plate;
 - direction - Direction of vertical movement of the car in the frame. An optional field, may be absent. If present, it can take the values “up” or “down”.

Getting a list of employees

5. Sigur ACS initiates a GET request: `/getpersons` to synchronize data with Access. The request is executed when the ACS server starts and when any synchronized data on the ACS side changes. If the request or subsequent requests fail, the ACS will try to repeat the request every 5 seconds until synchronization is completed without errors.
6. Access returns a json object with the fields:
 - id - employee ID;
 - photoVersion - Current “photo version” of the employee in Access, the field value is the time the photo was updated in the ACS in timestamp format (unix format);
 - name - Employee name.

Employee modification

7. Sigur ACS initiates a POST request: `/updateperson` to add or update employee data in Access. The request body contains a json object with the following fields:
 - id - employee ID in ACS, an integer from 1 to $2^{31}-1$;

- name - employee name;
 - photoVersion - employee's current "photo version" in Access, the field value is the time the photo was updated in ACS in timestamp format (unix format);
 - photo - JPEG photo, encoded in base64.
8. Access searches its database for an employee with the specified id. If such an id exists, then his data must be updated to match the transferred data. If such an id does not exist, then it must be created. If successful, Access returns HTTP status 200. OK. If unsuccessful, the ACS will prematurely terminate data synchronization, i.e. will stop executing updateperson and removeperson requests, the ACS will make a new attempt in 5 seconds.
9. Sigur ACS initiates a GET request: /removeperson to delete employee data in Access. The request body contains a json object with the field:
- id - ID of the employee to be deleted.

Configuration of the passage direction

10. Access searches its database for an employee with the specified id and deletes their data. If successful, Access returns HTTP status 200. OK. If unsuccessful, the ACS will prematurely terminate data synchronization; the ACS will make a new attempt in 5 seconds.
11. The Sigur ACS initiates a GET request: /getchannels to identify channels of video surveillance devices created in Access. The request is executed when the user performs actions in the ACS interface to configure the connection of channels and passage points.
12. Access returns the json object "channels" with the fields:
- id - channel ID;
 - name - Channel name.

The connection of channels and passage points occurs in the ACS user interface.

17.6. Sigur FAQ

1. Why is the drop-down list in the **bio_system_id** field empty when configuring the Sigur service connection parameters?
 - It is necessary to check the presence of the added biometric system service, it must be of a supported type. Supported types: [Luna](#), [CbsMts](#).
2. Why is the drop-down list in the **luna_cars_id** field empty when configuring the Sigur service connection parameters?
 - It is necessary to check the presence of the added [LunaCars](#) service.
3. Why did employee synchronization not start after configuring Sigur?
 - The following points must be met in order for the data to synchronize:
 - Recognition is enabled in the ACS software settings;

- Recognition is enabled in the equipment. The access point settings must have a camera linked and the required mode (identification or verification) checked;
- The employee must have access to this access point;
- The employee must have an access mode in which the required identification mode is enabled;
- Any edit of any employee can be the trigger for starting employee synchronization.

18. STRAZH ACS

Performs replication of user data from the ACS to the specified list of the Biometric System and generates StrazhController controllers from the received list of devices for subsequent execution of requests for entry or exit.

- Supports the ACS version: 1.2.211201.648.

The integration supports operation in 1-phase and 2-phase modes.

18.1. Supported integration options for STRAZH ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

Transfer of user data from the ACS to LP5 occurs using two mechanisms:

- replication - the mechanism for the initial transfer of user data;
- synchronization - the mechanism for periodic transfer of user data when the composition/data of users changes.

For the synchronization/replication settings, see the service settings.

Each integration with LP5 (Table 68) uses the [Luna](#) service.

If the terminal does not have data output facilities (e.g., a screen), the [SendToDevice](#) pipeline is not required.

Table 68. LP5 integration options

Service	Device	Pipeline
1f		
Strazh + StrazhController	LunaFast4A1	MatchByPhoto + SendToDevice + SendToController
2f		
Strazh + StrazhController	Beward	Strazh2FA + MatchByPhoto
	BioSmart	Strazh2FA + MatchByPhoto
	Dahua	Strazh2FA + MatchByPhoto
	Dahua Thermo	Strazh2FA + MatchByPhoto
	Fortuna315	Strazh2FA + MatchByPhoto
	HikvisionCamera	Strazh2FA + MatchByPhoto

Service	Device	Pipeline
	HikvisionCamera Thermo	Strazh2FA + MatchByPhoto
	HikvisionTerminal Thermo	Strazh2FA + MatchByPhoto
	LunaFast4A1	Strazh2FA + MatchByPhoto
	Panda	Strazh2FA + MatchByPhoto
	UniUbi	Strazh2FA + MatchByPhoto
	VKVision02	Strazh2FA + MatchByPhoto
	R20Face	Strazh2FA + MatchByPhoto

In each integration with CBS (Table 69), the CBS service is used.

Table 69. KBS integration options

Service	Device	Pipeline
1f		
CbsMts + Strazh + StrazhController	Beward	MatchByPhoto + SendToDevice + SendToController
2f		
CbsMts + Strazh + StrazhController	Beward	MatchByPhoto + Strazh2FA
	Dahua	MatchByPhoto + Strazh2FA
	HikvisionCamera	MatchByPhoto + Strazh2FA
	LunaFast4A1	MatchByPhoto + Strazh2FA
	UniUbi	MatchByPhoto + Strazh2FA

18.2. Standard integration using STRAZH ACS

When integrating with STRAZH, standard Access components (Figure 106) and (Table 70) are used.

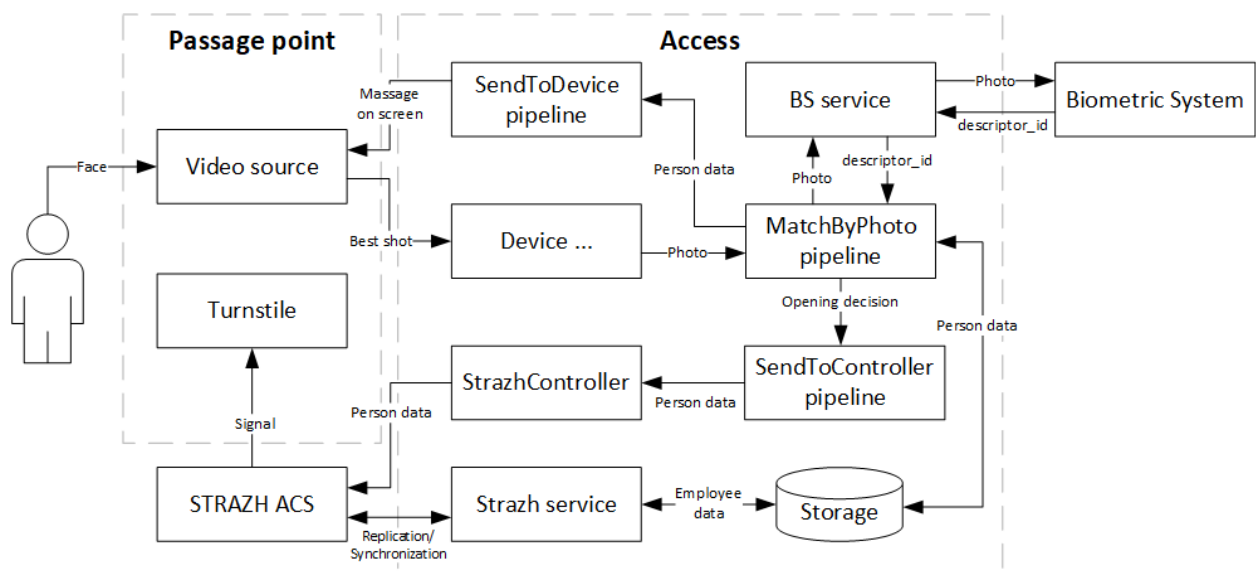


Figure 106. Component diagram for 1-phase integration

Table 70. Integration Description

Component	Description
Person	A person wishing to pass through a passage point.
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Video data source	A device for extracting a frame of a person's face. It can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream. A biometric terminal allows you to create feedback to show a person information about the passage.
Controller	Pass point control board.
Turnstile	Barrier device for access control
STRAZH ACS	Central software for working with Strazh. Stores employee data and makes decisions on granting access.
STRAZH Service	An Access component for replicating/synchronizing employees from the ACS and listening to ACS events.
StrazhController	Access Component for interacting with the ACS controller. A separate controller must be created for each reader.

Component	Description
Device ...	An Access component for receiving data from a video data source. Selected based on the device used.
MatchByPhoto Pipeline	Access Component for interaction with the BS. When working with a biometric terminal, it is necessary to additionally connect the SendToDevice pipeline
Biometric system	A system for comparing a reference photo of a person with the best frame obtained from a video data source. Can be either Luna or support KBS.
SendToController Pipeline	Access Component for sending card number and full name to StrazhController after matching a person and confirming the card number in Access.
Storage	Local system for storing relationships between ACS persons and their biometric data.

2f integration (Figure 107) and (Table 71).

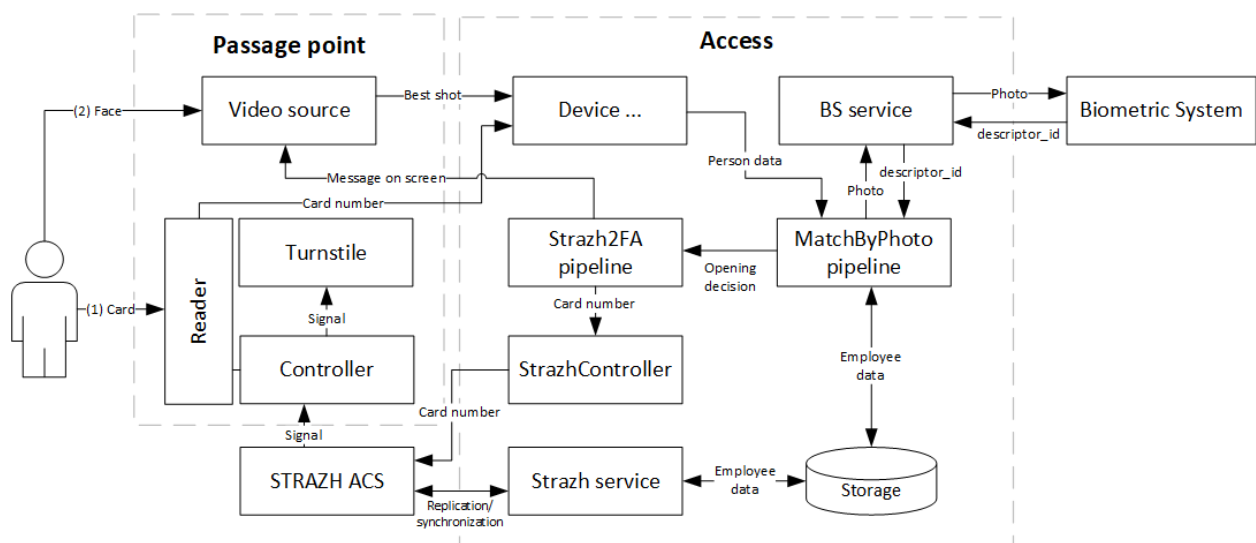


Figure 107. Component diagram for 2f integration

Table 71. Integration description

Component	Description
Person	A person wishing to pass through a passage point.

Component	Description
Passage point	A set of components used to control human access. More than one passage point can be connected, limited by the ACS license. A passage point can be used for both entry and exit. Each direction uses its own reader and video data source.
Video data source	A device for extracting a frame of a person's face. It can be either a biometric terminal (LUNA FAST 4A1 and others) or a camera connected via FaceStream. A biometric terminal allows you to create feedback to show a person information about the passage.
Controller	Passage point control board.
Device ...	An Access component for receiving data from a video data source. Selected based on the device used.
MatchByPhoto pipeline	Access component for interacting with the BS. When working with a biometric terminal, it is necessary to additionally connect the pipeline SendToDevice
BS Service	Access component for interaction with the Biometric System: for LP5 it is Luna , for KBS - the corresponding KBS service.
Biometric system	A system for comparing a reference photo of a person with the best frame obtained from a video data source.
Strazh service	Access component for replicating/synchronizing employees from the ACS and listening to ACS events.
Strazh2FA Pipeline	Access Component for Data Exchange with ACS
StrazhController	Access component for sending the card number to the ACS.
STRAZH ACS	Central software for working with the ACS. Stores employee data and makes decisions on granting access.
Turnstile	Barrier device for access control
Storage	Local system for storing relationships between ACS persons and their biometric data.

18.3. Setting up STRAZH ACS software for two-factor authentication

The ACS has implemented privilege levels. For 1F and 2F authorization to work, the access point privilege level must be equal to the employee's level, otherwise access will be denied.

If the employee has a higher privilege level than the access point, authorization will only occur by 1F (by card).

To configure the software, follow these steps:

1. Go to ACS Settings > Access Points > Create a new one (if not created).
2. Enter the required access point data if necessary.

This completes the process of setting up 1F authorization, follow steps 3-5 if 2F is planned.

3. Click the “Additional parameters” tab and add “Confirmation of access by an external system” with the value “Yes”.
4. Add the “Maximum waiting time for confirmation of access by an external system” parameter.
5. Adjust the timeout for waiting for a response from the external system and the default decision if the system does not have time to process the request.

After this, when trying to pass through this point with a card whose privilege level is lower than the privilege level of the point, an event with type: `access_confirmation` and data in the form of a JSON object with the fields `request` and `response` will be sent via the SSE mechanism.

The request contains a request for access, and the response contains a preliminary decision of the ACS about the possibility of access (i.e., a decision after standard checks of the profile, schedule, etc.).

Then the ACS waits for a decision on access to be sent to it via HTTP POST to `/access_confirmation` indicating the UUID of the request and a decision to let in or not.

Regardless of the ACS decision in response, the external system can let in or not let in.

18.4. Methods of interaction with STRAZH

An API is used to exchange data with the ACS (Table 72).

Table 72. STRAZH methods

Task	Method	Description
Log in	POST <code>/api/v1/login/</code>	Access authorization in ACS. Authorization occurs when adding a service and after the token expires
Get information about ACS	GET <code>/api/v1/info</code>	Getting the ACS version to check compatibility and display in the UI.
Get employees	GET <code>/api/v1/staff</code>	Replication and synchronization of employees (<code>person_id</code> , full name, photo) from ACS to local storage

Task	Method	Description
Get information about an employee	GET /api/v1/staff/{person_id}	Getting employee data from ACS (full name, photo)
Get employee photo	GET /api/v1/images/{person_id}	Get employee photo from ACS to send to biometric system
Get access points	GET /api/v1/access_points	Get access point (controller) IDs for manual matching of cameras/terminals and access points
Open access point	POST /api/v1/request_access	Send signal to controller to grant access.
Confirm 2nd factor	POST /api/v1/access_confirmation	Send confirmation to ACS about passing verification by the second factor (photo)
Open SSE connection	GET /sse	Open SSE connection to view event queue (reading, updating employee data)

18.5. STRAZH interaction process diagrams

18.5.1. Strazh service connection

Sequence diagram (Figure 108).

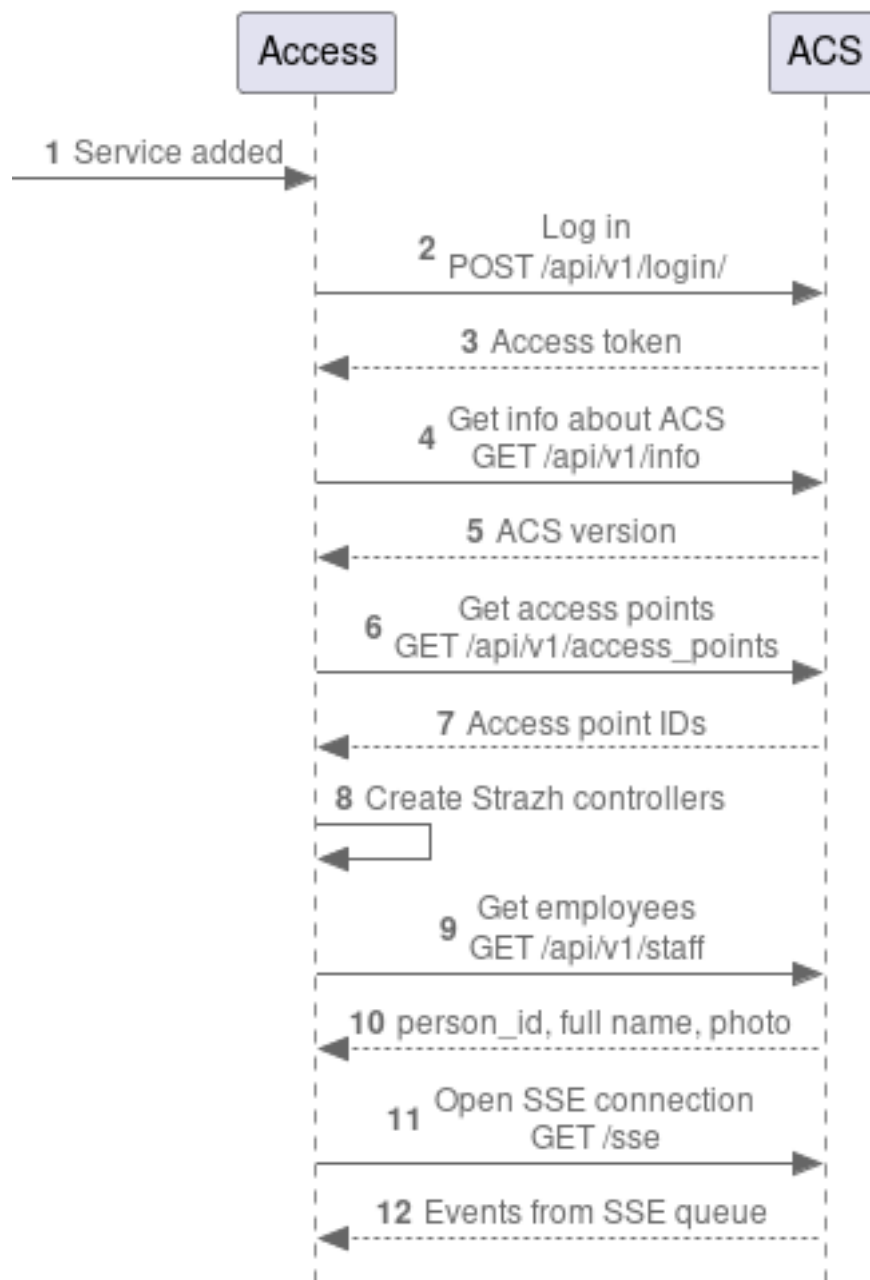


Figure 108. Process diagram for ACS connection

1. The user added the Strazh service to Access.
2. Access sends an authorization request to the ACS.
3. The ACS returns an authorization token. The token has a lifetime, after which Access re-performs authorization.
4. Access sends a request to obtain information about the ACS.
5. The ACS returns information. Access uses only the ACS version to check compatibility and user

information in the UI.

6. Access requests information about access points (controllers) connected to the ACS.
7. The ACS returns the access point IDs.
8. Access creates StrazhController controllers in accordance with the received IDs.
9. Access sends a request to obtain information about employees to replicate data to the local storage.
10. ACS returns person_id, full name and photo.
11. Access sends a request to open an SSE connection to view the list of events (changes in employees, access).
12. ACS opens an SSE connection with Access.

18.5.2. Modifying employees in STRAZH ACS

Sequence diagram (Figure 109).

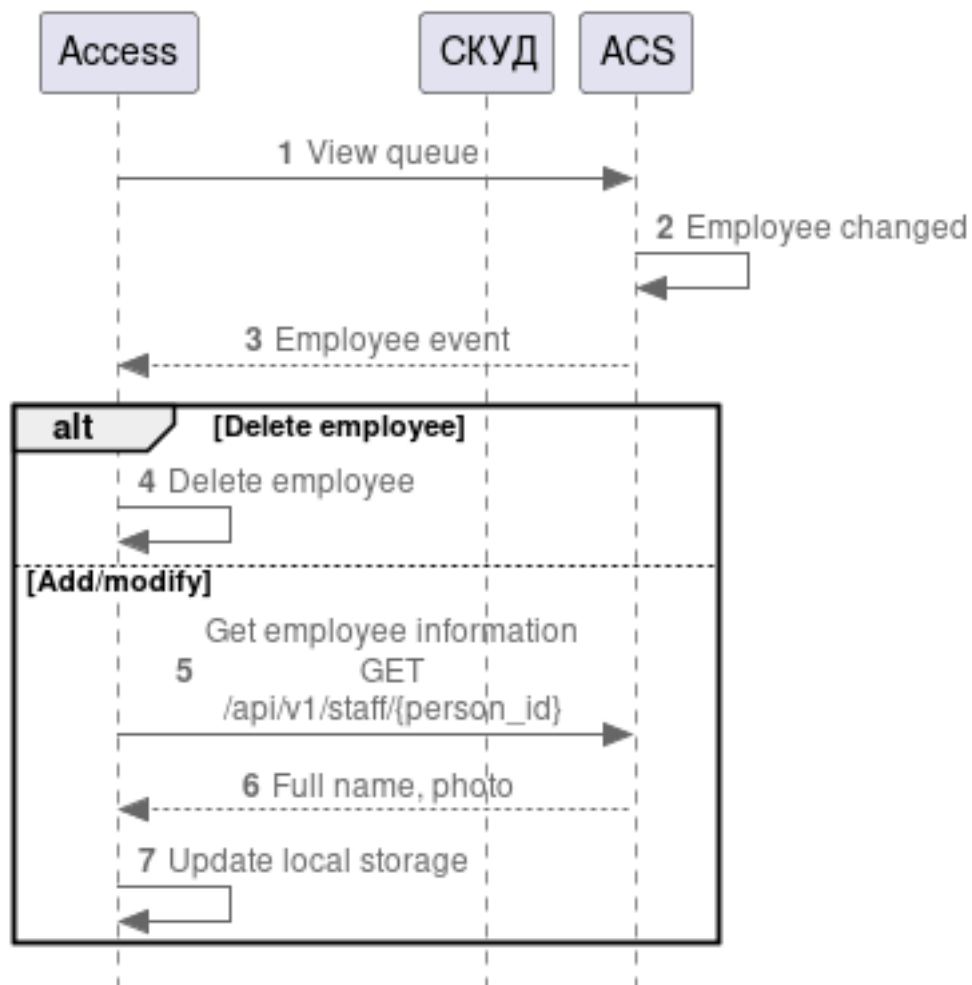


Figure 109. Process diagram for changing employees in ACS

1. Access views the event queue in ACS via SSE connection.
2. The employee is changed in ACS (added, changed or deleted).
3. Access finds events with the CREATE, MODIFY_DATA or DELETE tags in the queue.
4. Access deletes the employee from the local storage.
5. Access requests data on the employee by his person_id.
6. ACS returns the full name and photo of the employee.
7. Access updates the employee information in the local storage.

18.5.3. Processing STRAZH events with 1 factor

Sequence diagram (Figure 110).

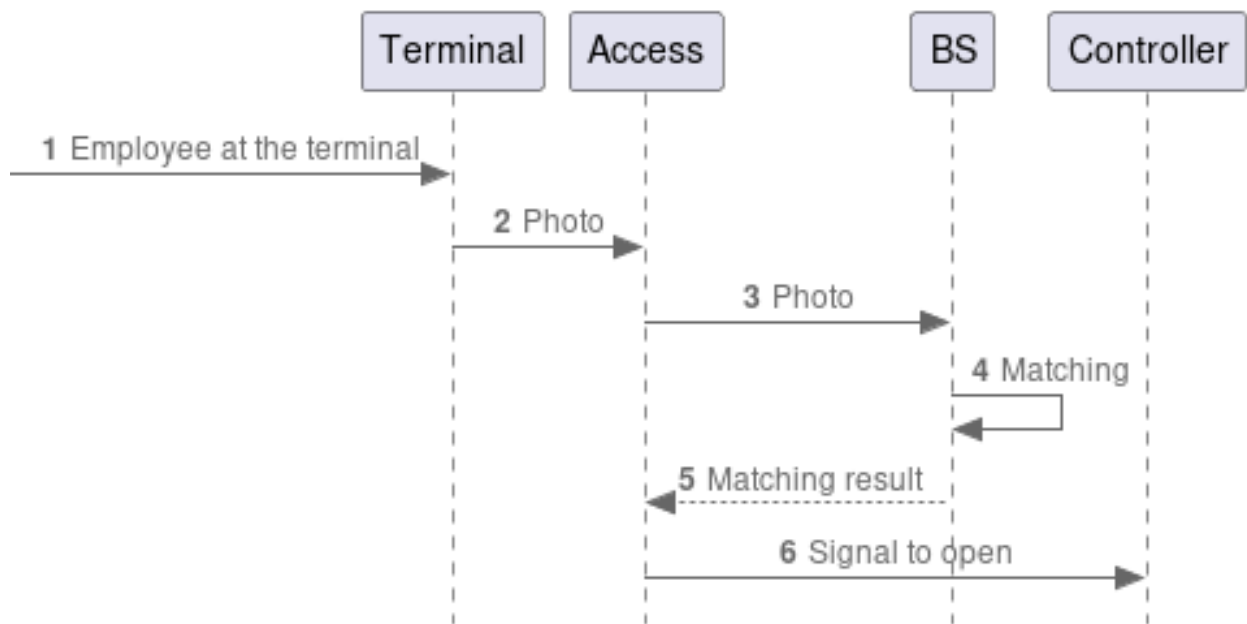


Figure 110. Process diagram with 1 factor

1. Employee at the biometric terminal at the checkpoint.
2. Terminal sends the best photo of the employee to Access.
3. Access sends a photo of the employee to the Biometric System.
4. BS compares the photo from the terminal and the one saved in the database.
5. BS returns to Access a decision on granting access.
6. Access sends a signal to the controller to open the access point.

18.5.4. Processing STRAZH events with 2 factors

Sequence diagram (Figure 111).

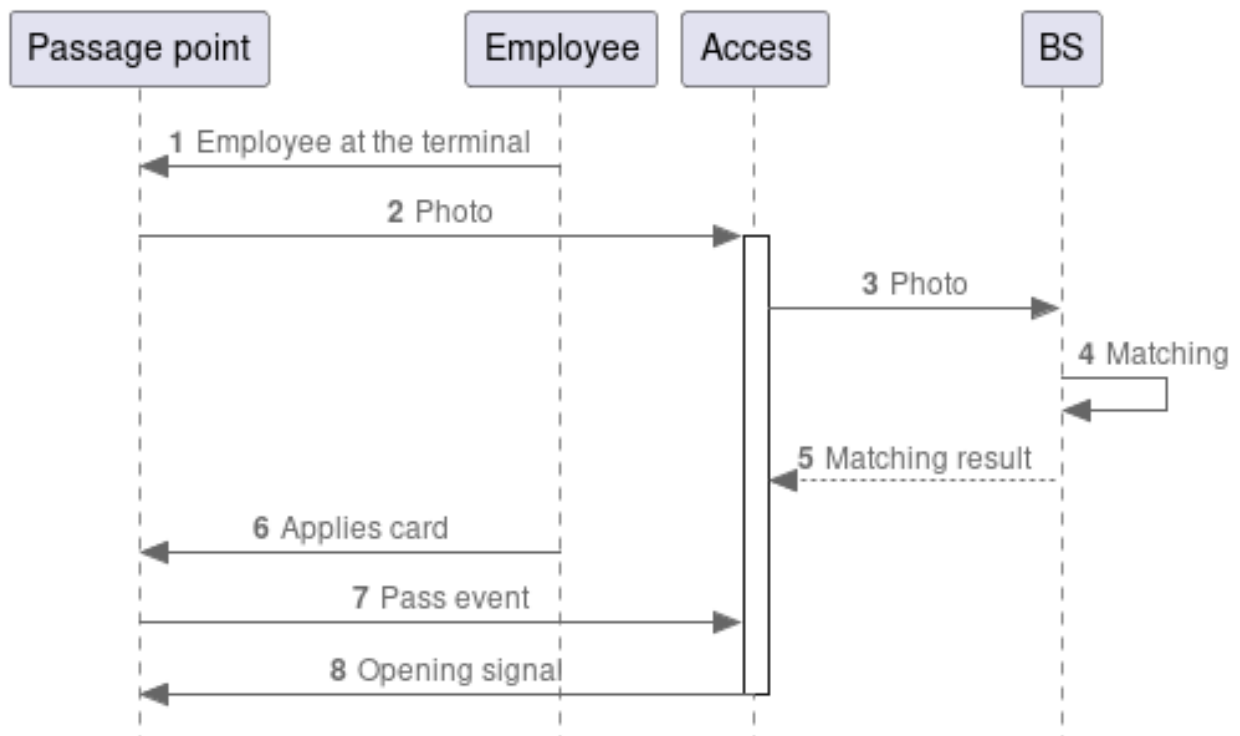


Figure 111. Process diagram with 2 factors

1. Employee at the biometric terminal at the checkpoint.
2. The terminal sends the best shot of the employee to Access.
3. Access sends the employee's photo to the Biometric System.
4. The BS compares the photo from the terminal and the one stored in the database.
5. The BS returns to Access a decision on granting access.
6. The employee applies the card (the card use subprocess does not depend on photo processing, but, as a rule, the photo arrives first).
7. The access point sends information to Access via SSE about the passage (access point ID, passage direction, and person_id).
8. Access aggregates information about each factor and sends a signal to the controller to open the access point.

19. Integrations without ACS

The face recognition device generates an event, Access passes the event to LP5, LP5 processes the event and returns the result to Access for further processing.

Each integration (Table 73) uses the [Luna](#) service.

Table 73. Integration options without ACS

Сервис	Устройство	Пайплайн
Without ACS	HikvisionRecognition OnBoard	SendToLuna
- (Bars-x)	LunaFast4A1	SendToBars + LunaEventListener + SendToLuna
LunaAceConverter	-	-

20. Controllers

Controllers are required to work with controllers from different manufacturers for communication between VisionLabs systems and access control devices from other manufacturers.

All fields are required unless otherwise stated in the description.

20.1. ApacsController

The Apacs controller is generated automatically when the Apacs service is running from the received pass points. Up to 4 readers are supported.

20.1.1. Setting up parameters for connecting to the Apacs controller

Apacs controller settings and possible values (Table 74):

Table 74. Apacs Controller Settings

Parameter	Description	Possible values	Default value
name	User-defined device name	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
group	Name of the list for grouping components	Any numeric and text values	-
apacs_id	ID of the Apacs instance in Access	-	-
external_-controller_id	The unique identifier of the device used in the integration. Specified in Access.	Device UUID	-
retry_entry_-sleep_interval	The pause interval between the passage in seconds, the larger the flow of people, the more there should be a pause.	1 ...10	5

Parameter	Description	Possible values	Default value
(1-8)_source	Event source (camera or terminal) linked to the corresponding reader	-	-

20.2. Gate controller

The GateController controller is designed to work with the GateEthernetWiegand converter, which can be used to send the Wiegand format card number to the controller. To run the controller, specify IP, port, and component identifiers to the corresponding device outputs in order to understand which direction to open when receiving detections from devices.

20.2.1. Gate Ethernet-Wiegand converter

A specialized Gate-Ethernet/Wiegand interface converter (version 2) is used to connect recognition servers (face, vehicle registration plate or other identification feature) to the ACS controller using a special protocol. The converter provides reception of a code message via the Ethernet network from the recognition server, decoding of the received message and issuance of an identifier code to the required Wiegand input of the ACS controller. The converter is configured using a special utility — a program running under the Windows operating system. The program sets the initial IP address of the device and other communication Parameters.

20.2.2. Gate controller settings

Gate Controller settings and possible values (Table 75):

Table 75. Gate Controller Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the server where Gate is installed	IP address in the form of X.X.X.X or site.domain	-

Parameter	Description	Values	Default value
port	Port of the server where Gate is deployed	-	5000
entry_source	The name of the event source, expected as the source of the event bound to the card reader 0	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-
exit_source	The name of the event source, expected as the source of the event bound to the card reader 1	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-
retry_entry_sleep_interval	Pause interval between passages (in seconds).The greater the flow of people, the longer the pause should be	1...10	7

20.3. LaurentController

The Laurent2 controller is designed to control and manage access together with the Luna Cars service.

- Supported devices: Laurent2.
- Supported firmware versions: L212.

20.3.1. Laurent controller settings

Controller settings and possible values (Table 76):

Table 76. Laurent Controller Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the device	IP address in the form of X.X.X.X or site.domain	-
port	Port of the device	-	-
password	Controller password	-	-4
enable_ssl	Method of encrypting data during transmission over the network. Depends on the network type in the solution.	On - https	Off
		Off - http	
delay_time	Relay pen time (sec)	1...10	-

20.4. PercoController

The PERCo controller is generated automatically during the operation of the PERCo-Web service from the devices received at startup. To use it, you must manually enter the `entry_source` and `exit_source` values for each of the created controllers.

20.4.1. PercooController settings

Controller settings and possible values (Table 77):

Table 77. PERCo Controller Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
perco_web_id	PercoWEB service instance ID in Access	-	-
external_controller_id	The unique identifier of the device used in the integration. Specified in Access.	Device UUID	-
description	Additional field for entering a description of the passage point	Russian and Latin characters are supported. It is not recommended to enter more than 50 characters	-
entry_source	The name of the event source, expected as the source of the event bound to the card reader 1	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-
exit_source	The name of the event source, expected as the source of the event bound to the card reader 2	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-
retry_entry_sleep_interval	The pause interval between the passage in seconds, the larger the flow of people, the more there should be a pause.	1...10/5	

20.5. PusrController

The controller is designed to work with the WGNetConverter converter, which can be used to send the Wiegand format card number to the controller.

- Supported devices: WG-TCP
- Supported firmware: V6005

It is important that the converter is in TCP Server mode. To do this, during the initial setup, select the appropriate Work Mode in the web interface in the Serial Port section.

20.5.1. Pusr controller settings

Controller settings and possible values (Table 78):

Table 78. Settings of the Pusr Controller

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the server where Pusr is installed	IP address in the form of X.X.X.X or site.domain	-
port	Port of the server where Pusr is deployed	-	-
entry_source	The name of the event source, expected as the source of the event bound to the card reader 1	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-
exit_source	The name of the event source, expected as the source of the event bound to the card reader 2	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-

Parameter	Description	Values	Default value
retry_entry_sleep_interval	Pause interval between passages (in seconds).The greater the flow of people, the longer the pause should be	1...10	5

20.6. Salto controller

The Salto controller is generated automatically when the Salto service is running from the received pass points. For use and operation, after generation, it is necessary to manually enter the `entry_source` value for each of the created controllers.

20.6.1. Salto controller settings

Controller settings and possible values (Table 79):

Table 79. Salto Controller Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
salto_id	Salto service instance ID in Access	-	-
external_controller_id	The unique identifier of the device used in the integration. Specified in Access.	Device UUID	-

Parameter	Description	Values	Default value
entry_source	Drop-down list for selecting the event source, expected as a source event bound to the door	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-
retry_entry_sleep_interval	Pause interval between passages (in seconds).The greater the flow of people, the longer the pause should be	1...10	5

20.7. Strazh controller

The Strazh controller is generated automatically when the Strazh service is running from the devices received at startup. After generation, it is necessary to manually enter the source value for each of the created controllers.

20.7.1. Strazh controller settings

Controller settings and possible values (Table 80):

Table 80. Strazh Controller Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
retry_entry_sleep_interval	The pause interval between the passage in seconds, the larger the flow of people, the more there should be a pause.	1 ...10	5

Parameter	Description	Values	Default value
external_controller_id	The unique identifier of the device used in the integration. Specified in Access.	Device UUID	-
strazh_id	Strazh service instance ID in Access	-	-
second_factor_expiry_time	Time limit in seconds to receive the second factor when using two-factor authentication. It is not recommended to set the limit to more than 10 seconds.	0...10	-
entry_source	The name of the event source, expected as the source of the event bound to the card reader A	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-
exit_source	The name of the event source, expected as the source of the event bound to the card reader B	Any textual names. Latin and Cyrillic input is supported. It is not recommended to enter more than 30 characters.	-

21. Devices

To perform hardware-software integration of LP5/LUNA CARS for access control, it is necessary to use devices - a terminal, cameras, etc.

All fields are required unless otherwise stated in the description.

21.1. Beward

The biometric terminal with temperature measurement, mask detection, and built-in relay.

- Supported devices: TFR80-210T1Q / TFR80-210.
- Supported firmware versions: 1.2.13.0 / 2.1.6.0.

21.1.1. Beward settings

Device settings and possible values (Table 81):

Table 81. Settings of Beware

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
successful_pass_message_template	Message on successful identification. To display the username on the terminal screen upon successful identification, you must use the full name variables. The order of words in the greeting message can be any.	Any text names and Full name variables . Only Latin characters are supported. It is not recommended to enter more than 33 characters	Welcome!

Parameter	Description	Values	Default value
successful_pass_message_template	Message upon successful identification. To display the user's name on the terminal screen, upon successful identification, you must use the NAME variables. The word order in the welcome message can be any.	Any text names and full NAME variables . Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 24 characters.	-
group	List name for grouping components	Any numeric or text values	-
display_message_time_sec	The time for displaying text on the screen in seconds. It only works towards increasing the standard value set in the firmware.	Integers greater than zero	-
host	IP address of the server where Beward is installed	IP address in the form of X.X.X.X or site.domain	-
port	The port of the server where Beware is deployed	-	80
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled;	Off
		Off — disabled	
login	Beward user login. Input of Latin characters, numbers and symbols is supported.	The user created in Beward	-
password	Beward user password. Input of Latin characters, numbers and symbols is supported.	User password	-

Parameter	Description	Values	Default value
open_door_time	Relay closing time in milliseconds	The time is taken from the relay manual	2000
vl_access_host	IP address of the server where Access is installed	IP address in the form of X.X.X.X or site.domain	-
vl_access_port	Port of the server where Access is deployed	-	9091
event_expiry_time	Event validity time (in seconds). It is necessary to reduce the time, with a large flow of people, as the device cache may overflow	>10	60
handler_id	UUID of the passage event handler, created in Luna	-	-
time_change_interval	The frequency of time updates on the device is set in minutes. It is recommended to use it if the time on the terminal is lost.	Integers greater than zero	60

21.2. BioSmart Quasar

- Supported devices: BioSmart Quasar.
- Supported firmware versions: 2.3.0.46.

21.2.1. BioSmart Quasar settings

Settings for creating a new device (Table 82):

Table 82. BioSmart Quasar Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
handler_id	UUID of the passage event handler, created in Luna	-	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the server where Quasar is installed	IP address in the form of X.X.X.X or site.domain	-
port	Port of the server where Quasar is deployed	-	80
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off
luna_id	Drop-down list with Luna service IDs in Access	-	-
min_face_similarity	Minimum face similarity threshold when performing verification.	The value is formed at the design stage and corrected at the testing stage.	0.00...1.00

To subscribe to events, go to “Settings” on the terminal → go to “Server Identification” → select the server type: BioSmartLite → enter the endpoint for sending data: `http://server_IP/vl-access/webhook/biosmart/` → save the settings.

The device does not generate events and does not put anything in the queue. Requests to Luna are sent directly from endpoints.

21.3. Dahua

Some models of Dahua cameras have a relay and the ability to control it programmatically.

During the implementation of the project, LP5 is integrated with this functionality, which allows to control the relay when a face from a certain list appears in the frame.

For example, it is possible to send a signal to an electronic door lock so that the door opens or does not open.

The device starts a stream connection, generates, and puts a face detection event in the queue.

21.3.1. Dahua settings

Device settings and possible values (Table 83):

Table 83. Dahua Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
device_id	Internal ID of the device. Specified in the device settings	-	-
host	Dahua camera IP address	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to Dahua camera	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off

Parameter	Description	Values	Default value
login	Dahua user login. Input of Latin characters, numbers and symbols is supported.	The user created in Beward	-
password	Dahua user password. Input of Latin characters, numbers and symbols is supported.	User password	-

21.4. DahuaThermo

Some models of Dahua cameras have a relay and the ability to control it programmatically.

During the implementation of the project, LP5 is integrated with this functionality, which allows to control the relay when a face from a certain list appears in the frame.

For example, it is possible to send a signal to an electronic door lock so that the door opens or does not open.

- Supported system version: 2.631.0000000.31.T, build date 2020-07-06.

The device starts an HTTP stream connection to the thermal imaging camera and captures faces by sending a thermal detection event of the face to the queue.

21.4.1. DahuaThermo settings

To start, you must specify the following settings (Table 84):

Table 84. DahuaThermo Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
handler_id	UUID of the passage event handler, created in Luna	-	-

Parameter	Description	Values	Default value
group	List name for grouping components	Any numeric or text values	-
host	IP address of Dahua Thermo camera	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to Dahua Thermo camera	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off
login	Dahua Thermo user login. Input of Latin characters, numbers and symbols is supported.	The user created in Beward	-
password	Dahua Thermo user password. Input of Latin characters, numbers and symbols is supported.	User password	-
timeout	Timeout for an unsuccessful attempt to connect to the service. It is necessary to increase the time if there is a large delay between servers	The time is selected based on the delay in the network to maintain performance	-

21.5. Fortuna315

Generates Thermo events in the SendThermalEventToLuna queue based on the received data from devices. Includes paired devices — thermal imaging camera and camera.

Supported firmware versions of the camera: V4.02.00, the camera thermal imaging camera: 2.20.0.0.R26130.alpha8 V1.0. Supported hardware versions: V1.0. Supported algorithm versions:

smart2.0.0-06-2020.06.17.16:06:42.

21.5.1. Fortuna315 settings

To subscribe to events, you must create a device with the following settings (Table 85):

Table 85. Fortuna315 Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
handler_id	UUID of the passage event handler, created in Luna	-	-
group	List name for grouping components	Any numeric or text values	-
device_id	Internal ID of the device. Specified in the device settings	-	-
host	IP address of Fortuna315 camera	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to Fortuna315 camera	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off
thermo_host	IP address of the Fortuna315 thermal imaging camera	IP address in the form of X.X.X.X or site.domain	-
thermo_port	Port of the Fortuna315 thermal imaging camera	-	-

21.6. GrgFaster

The GrgFaster terminal has the ability to display a message on the screen and send the card number to the connected controller.

- Supported devices: GRG Banking Faster.
- Supported models: SV-M082f-C2.
- Supported firmware versions (FW): 1.004.30.3bb324.R.
- Supported Hardware Versions (HW): 1.0.0

21.6.1. Configuring settings for connecting to GrgFaster

Device settings and possible values (Table 86):

Table 86. GrgFaster Settings

Parameter	Description	Possible values	Default value
name	User-defined device name	Any text names. Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 30 characters.	-
group	Name of the list for grouping components	Any numeric and text values	-
host	IP address or domain name of the server with GrgFaster installed	IP address in the form of X.X.X.X. or site.domain.	-
port	The port of the server where GrgFaster is deployed	-	9166
enable_ssl	Supports SSL encryption for messages. It must be activated if necessary to maintain confidentiality. When activated, the load on the device and the message transmission time increases	On – active Off – inactive	Off

Parameter	Description	Possible values	Default value
login	Username of the GrgFasterd user. Input of Latin letters, numbers, and symbols is supported.	User created in GrgFaster	-
password	The password of the GrgFaster user. Input of Latin letters, numbers, and symbols is supported.	User's password	-
successful_pass_-message_-template	Message upon successful identification. To display the user's name on the terminal screen, upon successful identification, you must use the NAME variables. The word order in the welcome message can be any.	Any text names and full NAME variables . Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 35 characters.	-

21.7. HikvisionCamera

A camera for generating a video stream for LP5 with subsequent integration with ACS.

- Supported devices: DS-2CD3126G2-IS.
- Supported firmware versions: V5.5.134 build 200430.

The device generates events of type `FaceDetectionEvent`.

21.7.1. HikvisionCamera settings

To subscribe to events, create a device with the following settings (Table 87):

Table 87. HikvisionCamera Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
handler_id	UUID of the passage event handler, created in Luna	-	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the HikvisionCamera camera	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to HikvisionCamera camera	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled;	Off
		Off — disabled	
login	HikvisionCamera user login. Input of Latin characters, numbers and symbols is supported.	User created to access the device	-
password	HikvisionCamera user password. Input of Latin characters, numbers and symbols is supported.	User password	-
timeout	Timeout for an unsuccessful attempt to connect to the service. It is necessary to increase the time if there is a large delay between servers	The time is selected based on the delay in the network to maintain performance	10

21.8. HikvisionCameraThermo

A camera with temperature measurement and data transfer functions in LP5.

- Supported devices: DS-2CD3126G2-IS.
- Supported firmware versions: V5.5.134 build 200430.

Events in the queue are of type `ThermalEvent`.

21.8.1. HikvisionCameraThermo settings

To subscribe to events, create a device with the following settings (Table 88):

Table 88. HikvisionCameraThermo settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
handler_id	UUID of the passage event handler, created in Luna	-	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the HikvisionCameraThermo camera	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to HikvisionCameraThermo camera	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled;	Off
		Off — disabled	

Parameter	Description	Values	Default value
login	HikvisionCameraThermo user login. Input of Latin characters, numbers and symbols is supported.	User created to access the device	-
password	HikvisionCameraThermo user password. Input of Latin characters, numbers and symbols is supported.	User password	-
timeout	Timeout for an unsuccessful attempt to connect to the service. It is necessary to increase the time if there is a large delay between servers	The time is selected based on the delay in the network to maintain performance	10

21.9. HikvisionRecognitionOnBoard terminal

A biometric terminal with face recognition function.

- Supported devices: DS-K1T341AMF, DS-K1T341AM, DS-K1T680D-E1.
- Supported firmware versions: V3.2.30 build 220210.

Open the device's web interface, go to "Configuration" → "Access Control" → "Face Recognition Parameters" → "Working Mode" and make sure that the "Permission Free Mode" mode is set.

After adding, the faces from the specified list will be replicated to the device's memory. You can add/remove faces in Luna, the changes will be automatically applied to the device.

- the following format of the user_data field is required:

```
name;card number
```

- do not set or change the external_id field.

Updating of face data during editing is not supported. If it is necessary to update the data of a person, delete this person and add it again with the necessary data.

Events in the queue are of type FaceDetectionEvent.

21.9.1. HikvisionRecognitionOnBoard settings

To subscribe to events, create a device with the following settings (Table 89):

Table 89. HikvisionRecognitionOnBoard settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the HikvisionRecognitionOnBoard camera	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to HikvisionRecognitionOnBoard camera	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off
login	HikvisionRecognitionOnBoard user login. Input of Latin characters, numbers and symbols is supported.	User created to access the device	-
password	HikvisionRecognitionOnBoard user password. Input of Latin characters, numbers and symbols is supported.	User password	-
vl_access_host	IP address of the server where Access is installed	IP address in the form of X.X.X.X or site.domain	-

Parameter	Description	Values	Default value
vl_access_port	Port of the server where Access is deployed	-	9091
luna_id	Drop-down list with Luna service IDs in Access	-	-
recognition_interval	Interval between the start of recognition. It is set depending on the flow of people	1...10	1
liveness_level	Degree of checking the level of liveness	low— fast processing speed, low accuracy is reduced;	low
		middle — average processing speed and recognition accuracy;	
		high — accurate definition, increased resource consumption	
event_expiry_time	Event validity time (in seconds). It is necessary to reduce the time, with a large flow of people, as the device cache may overflow	>10	60
time_change_interval	The frequency of time updates on the device is set in minutes. It is recommended to use it if the time on the terminal is lost.	Integers greater than zero	60

21.10. HikvisionTerminalThermo terminal

The biometric terminal with temperature measurement, mask detection, and built-in relay.

Only events of the `AccessControllerEvent` type (having a measured temperature) will be processed, events of this type come from the terminal.

Events in the queue are of type `ThermalEvent`.

Hikvision terminal with temperature measurement function.

- Supported devices: DS-K1TA70MI-T, DS-K1T671TM-3XF, DS-K5671-3XF/ZU.
- Supported firmware versions: V3.2.32 build 210525.

21.10.1. HikvisionTerminalThermo settings

To subscribe to events, create a device with the following settings (Table 90):

Table 90. HikvisionTerminalThermo settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
successful_pass_message	Message on successful identificationTo display the username on the terminal screen upon successful identification, you must use the special flag. The order of words in the greeting message can be any. Any text names.	Any text names and Full name variables . Only Latin characters are supported. It is not recommended to enter more than 50 characters.	Welcome
unsuccessful_pass_message_template	Message on unsuccessful identification. To display the username on the terminal screen upon successful identification, you must use the full name variables. The order of words in the greeting message can be any.	It is not recommended to enter more than 50 characters	Face is not identified

Parameter	Description	Values	Default value
display_message_time_sec	The time for displaying text on the screen in seconds. It only works towards increasing the standard value set in the firmware.	Integers greater than zero	-
host	IP address of the HikvisionTerminalThermo camera	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to HikvisionTerminalThermo camera	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled;	Off
		Off — disabled	
login	HikvisionTerminalThermo user login. Input of Latin characters, numbers and symbols is supported.	User created to access the device	-
password	HikvisionTerminalThermo user password. Input of Latin characters, numbers and symbols is supported.	User password	-
event_expiry_time	Event validity time (in seconds). It is necessary to reduce the time, with a large flow of people, as the device cache may overflow	>10	60
card_recognition_interval	Interval between card recognitions	0...10	3

Parameter	Description	Values	Default value
face_recognition_interval	Interval between recognizing images with faces	1...10	3
liveness	Responsible for activating the liveness	On – active Off – inactive	On system
liveness_level	Degree of checking the level of liveness	low— fast processing speed, accuracy is reduced; middle — average processing speed and recognition accuracy; high — accurate definition, increased resource consumption	low
attempts_check_liveness	Number of attempts to pass the liveness check. It is necessary to increase the number of verification attempts at complex angles and shooting conditions in order to avoid false positive recognitions	5...15	10
vl_access_host	IP address of the server where Access is installed	IP address in the form of X.X.X.X or site.domain	-
vl_access_port	Port of the server where Access is deployed	-	9091
light_brightness_led	Brightness level of the LED backlight. The darker the area in front of the terminal, the brighter the backlight should be	0...100	50

Parameter	Description	Values	Default value
light_brightness_ir	Brightness level of the infrared (IR) illumination. The darker the area in front of the terminal, the brighter the illumination should be	0...100	50
voice_prompt	Terminal voice prompts about pass events or errors. To set up the prompts, see the official documentation of the terminal.	On/Off	Off
handler_id	UUID of the passage event handler, created in Luna	-	-
time_change_interval	The frequency of time updates on the device is set in minutes. It is recommended to use it if the time on the terminal is lost.	Integers greater than zero	60
clear_old_events_interval	Frequency of deleting old events (in seconds), to prevent terminal memory overflow	300...1200	600
wiegand_direction	Direction of wiegand operation	input - receive card from reader output - send card to controller	input

21.11. LunaFast2NextGen

21.11.1. Configuring parameters for connecting to LunaFast2NextGen

To subscribe to events, you must create a device with the following settings (Table 91):

Table 91. LunaFast2NextGen settings

Parameter	Description	Possible values	Default value
name	User-defined device name	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
group	List name for grouping components	Any numeric or text values	-
successful_pass_message_template	Message upon successful identification. To display the user name on the terminal screen upon successful identification, you must use the full name variables. The word order in the welcome message can be any.	Any text names and full name variables . It is not recommended to enter more than 33 characters.	Welcome
Example with full name output: Welcome, {fullname}.			
unsuccessful_pass_message	Message upon unsuccessful identification	Any text names. Both Latin and Cyrillic characters are supported. It is not recommended to enter more than 33 characters.	Face not identified
display_message_time_sec	Text display time on the screen in seconds. Works only in the direction of increasing the standard value set in the firmware.	Integer numbers greater than zero	-
host	IP address or domain name of the LunaFast4A1 camera	IP address in the form X.X.X.X. or site.domain.	-
port	Port for connecting to the LunaFast4A1 terminal	-	80

Parameter	Description	Possible values	Default value
enable_ssl	SSL encryption support for messages. Must be activated if privacy is required. When activated, the load on the device and the message transmission time increase	On - active Off - inactive	Off
face_recognition_interval	Interval between face image recognitions.	1...10	3
vl_access_host	IP address of the server where Access is installed	IP address in the form X.X.X.X. or site.domain	-
vl_access_port	Port of the server where Access is deployed	-	9091
event_receiving_mode	Mode for receiving events from LP5 (from version 5.53.0).	rtsp - protocol using a persistent connection webhook - callbacks via HTTP. Client - Luna Platform, server - Luna Access	webhook

21.12. LunaFast4A1

The biometric terminal with recognition function.

- Supported devices: DS-K1T341CMF, DS-K1T680D-E1, DS-K1T341AMF, DS-K1T341AM, VL LUNA FAST 4A1, VL LUNA FAST 8A1, 671, DS-K1T671M, ACT-T1341M, DS-K1T680DF-E1, DS-K5671-ZU.
- Supported firmware versions: V3.3.40 build 250106, V3.2.30 build 210415, V3.2.30 build 210525, V3.2.30 build 210526, V3.2.30 build 210812, V3.2.30 build 211025, V3.2.30 build 220607, V3.2.30 build 220803, V3.2.30 build 221027, V3.2.33 build 210816, V3.2.35 build 220415, V3.2.35 build 220817.

Events in the queue are of type `FaceDetectionEvent`.

To disable the output of the greeting on the terminal screen, you must disable the `LunaEventListener` pipeline.

Can send card number to [controller](#) via [device](#).

The status of card number sending functionality support is displayed in the “info” block in the `hardware_with_card_sending` parameter after the component is connected.

21.12.1. LunaFast4A1 settings

To subscribe to events, create a device with the following settings (Table 92):

Table 92. LunaFast4A1 Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
successful_pass_message	Message on successful identificationTo display the username on the terminal screen upon successful identification. The order of words in the greeting message can be any. Any text names.	Any text names and Full name variables . It is not recommended to enter more than 50 characters.	Welcome
unsuccessful_pass_message_template	Message on unsuccessful identification. To display the username on the terminal screen upon successful identification, you must use the full name variables. The order of words in the greeting message can be any.	Only Latin characters are supported. It is not recommended to enter more than 50 characters	Face is not identified

Full name example:
Welcome, {fullname}.

Parameter	Description	Values	Default value
display_message_time_sec	The time for displaying text on the screen in seconds. It only works towards increasing the standard value set in the firmware.	Integers greater than zero	-
host	IP address of the LunaFast4A1 camera	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to the LunaFast4A1 terminal	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off
login	LunaFast4A1 user login. Input of Latin characters, numbers and symbols is supported.	User created to access the device	-
password	LunaFast4A1 user password. Input of Latin characters, numbers and symbols is supported.	User password	-
event_expiry_time	Event validity time (in seconds). It is necessary to reduce the time, with a large flow of people, as the device cache may overflow	>10	60
card_recognition_interval	Interval between card recognitions	0...10	3
face_recognition_interval	Interval between recognizing images with faces	1...10	3

Parameter	Description	Values	Default value
liveness	Responsible for activating the liveness	On – active	On system
		Off – inactive	
liveness_level	The degree of checking the level of liveness	low— fast processing speed, low accuracy is reduced;	low
		middle — average processing speed and recognition accuracy;	
		high — accurate definition, increased resource consumption	
attempts_check_liveness	Number of attempts to pass the liveness check. It is necessary to increase the number of verification attempts at complex angles and shooting conditions in order to avoid false positive recognitions	Values > 0	10
vl_access_host	IP address of the server where Access is installed	IP address in the form of X.X.X.X or site.domain	-
vl_access_port	Port of the server where Access is deployed	-	9091
light_brightness_led	Brightness level of the LED backlight. The darker the area in front of the terminal, the brighter the backlight should be	0...100	50
light_brightness_ir	Brightness level of the infrared (IR) illumination. The darker the area in front of the terminal, the brighter the illumination should be	0...100	50

Parameter	Description	Values	Default value
voice_prompt	Terminal voice prompts about pass events or errors. To set up the prompts, see the official documentation of the terminal.	On/Off	Off
handler_id	UUID of the passage event handler, created in Luna	-	-
time_change_interval	The frequency of time updates on the device is set in minutes. It is recommended to use it if the time on the terminal is lost.	Integers greater than zero	60
clear_old_events_interval	Frequency of deleting old events (in seconds), to prevent terminal memory overflow	300...1200	600
wiegand_direction	Direction of wiegand operation	input - receive card from reader output - send card to controller	input

21.13. Panda

A thermal imaging camera with face recognition.

- Supported devices: SN-T5/13, SN-F22-13.
- Supported firmware versions: v3.6.0825.1004.1.0.23.0.0, v3.6.0840.1004.1.45.1.0.2.

Events in the queue are of type `ThermalEvent`.

21.13.1. Panda settings

To subscribe to events, create a device with the following settings (Table 93):

Table 93. Panda Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
handler_id	UUID of the passage event handler, created in Luna	-	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the Panda camera	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to Panda camera	-	80
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled;	Off
		Off — disabled	

Go to the web interface of the device → sign in → go to the “Configuration” tab → select the “Network Service” section on the left panel → go to “CGI Alarm Service Center”.

Fill in the fields in the “CGIAlarm” section: assign endpoint as URL Start and URL End to send data to Access — `http://<vl_access_host>:<vl_access_port>/vl-access/webhook/device/<component_id>/event/handle_event/.` If necessary, in the “Proxy Settings” section, fill in the Address and Port fields: Access host and Access port, respectively. Save settings after configuring.

21.14. R20Face

The biometric terminal with protective mask detection and centralized control.

- Supported devices: R20-Face-T8.
- Supported firmware versions: GD-V32.7267.

21.14.1. R20Face settings

The following settings are used when creating a new device (Table 94):

Table 94. R20Face Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
successful_pass_message_template	Message on successful identification. To display the username on the terminal screen upon successful identification, you must use the full name variables. The order of words in the greeting message can be any.	Any text names and Full name variables . Only Latin characters are supported. It is not recommended to enter more than 33 characters	Welcome!
unsuccessful_pass_message	Message on unsuccessful identification. To display the username on the terminal screen upon successful identification, you must use the full name variables. The order of words in the greeting message can be any.	Any text names. Only Latin characters are supported. It is not recommended to enter more than 33 characters	Face is not identified
display_message_time_sec	The time for displaying text on the screen in seconds. It only works towards increasing the standard value set in the firmware.	Integers greater than zero	-
host	IP address of the server where R20Face is installed	IP address in the form of X.X.X.X or site.domain	-

Parameter	Description	Values	Default value
port	Port of the server where R20Face is deployed	-	8080
password	Terminal user password	User password	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off
vl_access_host	IP address of the server where Access is installed	IP address in the form of X.X.X.X or site.domain	-
vl_access_port	Port of the server where Access is deployed	-	9091
event_expiry_time	Event validity time (in seconds). It is necessary to reduce the time, with a large flow of people, as the device cache may overflow	>10	60
time_zone	Time zone	-12...+12	+3
handler_id	UUID of the passage event handler, created in Luna	-	-
handle_detection_events	Enabling the terminal detection event processing mode	On/Off	Off

21.15. UniUbi terminal

The biometric terminal with temperature measurement, mask detection, and built-in relay control functions.

- Supported devices: Uface 8-C temp, Uface 8T temp, R20-Face-T8.
- Supported firmware versions: GD-V30.7219, GD-V32.7247, GD-V32.7267.

When using a terminal without measuring the temperature, you need to change the `SentThermalEventToLuna` pipeline to `SendToLuna`.

21.15.1. UniUbi settings

To subscribe to events, create a device of the UniUbi type.

The following settings are used when creating a new device (Table 95):

Table 95. UniUbi Settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
handler_id	UUID of the passage event handler, created in Luna	-	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the UniUbi terminal	IP address in the form of X.X.X.X or site.domain	-
display_message_time_sec	The time for displaying text on the screen in seconds. It only works towards increasing the standard value set in the firmware.	Integers greater than zero	-
port	Port for connecting to the UniUbi terminal	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off
password	UniUbi user password	User password	-

Parameter	Description	Values	Default value
enabled_temp_mode	Temperature measurement mode. If disabled, change the <code>SentThermalEventToLuna</code> pipeline to <code>SendToLuna</code> .	On — enabled; Off — disabled	On
vl_access_host	IP address of the server where Access is installed	IP address in the form of X.X.X.X or site.domain	-
vl_access_port	Port of the server where Access is deployed	-	9091
event_expiry_time	Event validity time (in seconds). It is necessary to reduce the time, with a large flow of people, as the device cache may overflow	>10	60
time_zone	Time zone	-12...+12	+3
successful_pass_message_template	Message on successful identification. To display the username on the terminal screen upon successful identification, you must use the full name variables. The order of words in the greeting message can be any.	Any text names and Full name variables . Only Latin characters are supported. It is not recommended to enter more than 27 characters	Welcome
unsuccessful_pass_message	Message on unsuccessful identification	Any text names. Only Latin characters are supported. It is not recommended to enter more than 27 characters	Face is not identified

21.16. VKVision02

Terminal with functions of video recording and display of images on the screen.

- Supported devices: VANCOR VK VISION 02.

21.16.1. VKVision02 settings

The following settings are used when creating a new device (Table 96):

Table 96. VKVision02 settings

Parameter	Description	Values	Default value
name	Device name specified by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
group	List name for grouping components	Any numeric or text values	-
host	IP address of the VKVision01 terminal	IP address in the form of X.X.X.X or site.domain	-
port	Port for connecting to the VKVision01 terminal	-	-
enable_ssl	SSL encryption support for messages. Activate if privacy is required. When activated, the load on the device and the message transmission time increases	On — enabled; Off — disabled	Off
success_status_image_ttl	Display time (ms) of the image on the terminal with the status “Success” (in seconds)	5000...20000	10000
success_status_image_ttl	Display time (ms) of the image on the terminal with the status “Denied” (in seconds)	5000...20000	10000
luna_id	Drop-down list with Luna service IDs in Access	-	-

22. Pipelines

All fields are required unless otherwise stated in the description.

22.1. Apacs2FA

The Apacs2FA pipeline implements custom two-factor authentication for the Apacs service. Listening to events that occurred when reading the card, as well as face detection events. They are matched by their source. When the first factor arrives with a certain source, then the second factor starts waiting, the time for waiting for it is set in the expiry_time setting, as soon as it is received, the validation of this pair begins.

- The similarity of the best matching candidate must not be lower than that specified in the min_similarity setting.
- The card number received from the reader must match the card number of the person in the Luna list.

Upon completion of the authentication procedure, the corresponding text is displayed on the screen of the device. After successful validation, the card number is sent to the corresponding output of the Gate Ethernet Wiegand controller.

To create a pipeline, you need to specify (Table 97):

Table 97. Pipeline settings

Parameter	Description	Possible values	Default value
name	Pipeline name set by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters.	-
luna_id	Dropdown list to select the Luna Service ID in Access.	-	-
luna_id	Luna service identifier in Access	Dropdown list for service selection	-
apacs_id	Unique identifier of the Apacs service in Access	Dropdown list for service selection	-
min_face_similarity	Minimum face similarity threshold when performing verification.	The value is formed at the design stage and corrected at the testing stage.	0.00...1.00

Parameter	Description	Possible values	Default value
unknown_card_-message	Message when sending an unknown card to make an access decision.	Any text names. Only Latin characters are supported. It is not recommended to enter more than 50 characters.	The card has been read and sent to the controller
waiting_time_-exceeded_-message	Message when the second factor timed out	Any text names. Latin and Cyrillic input is supported. It is not recommended to enter more than 50 characters.	Timeout exceeded
access_denied_-card	Card number of the user created to notify about unsuccessful passes. You must specify the number so that in the connected ACS you can see the log that the authorization failed.	-	None
use_cards_-without_face	Using and sending to the controller cards that are not linked to faces	On - send Off - do not send	On

22.2. CreateBastionEvent

The pipeline works together with the Bastion ACS.

When single-factor authentication is enabled on the access point, it listens for ResultMatchEvent events and generates a BastionEvent event.

When the enable_negative_events mode is enabled, additionally sends a message to the terminal upon a negative matching event (required only for integration with **Bastion 3**).

If two-factor authentication is enabled, a request for access confirmation to the ACS is made.

The following settings are used when creating a new pipeline (Table 98):

Table 98. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name defined by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
bio_system_id	Selecting the LP5/CBS service name in Access.	-	-
bastion_id	Drop-down list for selecting the Bastion device ID used in the integration. Specified in Access.	-	-
enable_negative_events	Sends messages to the terminal when the matching result is below the min_face_similarity threshold (configured in the pipeline for receiving data from the biometric system). Only for Bastion 3	On - enable, Off - disable	Off

22.3. Custom2FA

The Custom2FA pipeline implements custom two-factor authentication. Card swiping events are listened to, as well as regular face detection events. Events are matched by their source and saved to the mapping. When the first factor arrives with a specific source, the waiting for the second factor starts. The time for waiting for it is set in the expiry_time setting, as soon as it is received, the validation of this pair begins:

- the similarity of the best matching candidate must not be lower than that specified in the min_face_similarity setting;
- the card number received from the reader must match the card number of the face in the Luna list.

Upon completion of the authentication procedure, the corresponding text is displayed on the device screen. After successful validation, the card number is sent to the appropriate output of the Gate Ethernet Wiegand or WGNetConverter controller.

The following settings are used when creating a new pipeline (Table 99):

Table 99. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name set by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters.	-
bio_system_id	Selecting the LP5/CBS service name in Access.	-	-
expiry_time	Time limit in seconds to get the second factor.	0...10	5
unknown_card_message	Message when sending an unknown card to decide on access	Any text names. Only Latin characters are supported. It is not recommended to enter more than 50 characters.	The card has been read and sent to the controller
waiting_time_exceeded_message	Message when the second factor timed out	Any text names. Latin and Cyrillic input is supported. It is not recommended to enter more than 50 characters.	Timeout exceeded
use_cards_without_face	Using and sending to the controller cards that are not linked to faces	On - send Off - do not send	On

22.4. LunaEventListener

LunaEventListener listens to events from Luna generated by Luna internal service, LunaStreams or any other external software. Sends maps to controller or converter and can send messages to device.

The following settings are used when creating a new pipeline (Table 100):

Table 100. Pipeline settings

Parameter	Description	Values	Default value
name	User-defined pipeline name	Any text names. Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 30 characters.	-
luna_id	Luna service ID in the system	-	-
min_face_similarity	Minimum face similarity threshold when performing verification.	The value is formed at the design stage and corrected at the testing stage.	0.00...1.00
enable_fake_events	Handling requests with failed Liveness checks to view such events. For activation, the enable_fake_events parameter in the SendToLuna pipeline must be active	On - handle	Off
		Off - do not process	

22.5. MatchByPhoto

Requests a descriptor in the CBS and extracts the candidate from its database using its identifier.

When working with devices and controllers, it is necessary to connect pipelines [SendToDevice](#) and [SendToController](#), respectively.

When creating a new pipeline, the following settings are used (Table 101):

Table 101. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-

Parameter	Description	Values	Default value
bio_system_id	Unique CBS service identifier	-	-
pacs_id	Unique service identifier in Access	Dropdown list for service selection	-
min_face_similarity	Minimum face similarity threshold when performing verification.	The value is formed at the design stage and corrected at the testing stage.	0.00...1.00
retry_entry_sleep_interval	Pause for reattempting the passage	>0 или None	5

22.6. MatchByPhotoInCbsAlpha

Requests a descriptor in CBS Alpha and extracts the candidate from its database using its identifier. With the received data (candidate and source name), an event is created and a SuccessMatchEvent is sent to the queue.

When working with devices and controllers, it is necessary to connect pipelines [SendToDevice](#) and [SendToController](#), respectively.

When creating a new pipeline, the following settings are used (Table 102):

Table 102. Pipeline settings.

Parameter	Description	Possible values	Default value
name	User-defined pipeline name	Any text names. Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 30 characters.	-
bio_system_id	The unique identifier of the CBS Alpha service	-	-
pacs_id	Unique service identifier in Access	Dropdown list for service selection	-
min_face_similarity	The minimum threshold for facial similarity during verification.	The value is formed at the design stage and adjusted at the testing stage.	0,00...1,00

Parameter	Description	Possible values	Default value
retry_entry_sleep_interval	Pause for reattempting the passage	1...10	5
only_cbs_list	Switching lists for matching	On - switching to work only with the KBS list Off - switching to work with two lists	Off

22.7. MatchInformerWebHook

Monitors events from a biometric system (LP5 or CBS), retrieves candidate information and sends it to an external service via a webhook. If no candidate is found, returns an empty `best_candidate` value.

The pipeline is required for solutions where Access needs to pass data from related components to external systems beyond the proposed [integrations](#).

The template of the returned data:

```
{
  source: str, best_candidate:
  {
    person_id: str, fullname: str | None, descriptor_id: str | None =
    None
  }
}
```

Description:

- source - device name;
- best_candidate - candidate data;
- person_id - candidate ID in ACS;
- fullname - candidate full name;
- descriptor_id - descriptor ID;

To create a pipeline, you need to specify (Table 103):

Table 103. Pipeline settings

Parameter	Description	Possible values	Default value
name	Pipeline name specified by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
host	IP address or domain name of the target server	IP address in the form X.X.X.X. or site.domain.	-
port	Target server port	-	-
urn	urn for connecting to an external service	-	-
enable_ssl	Data encryption method for network transmission. Depends on the network type in the solution.	On - https Off - http	Off

22.8. MatchInformerWebSocket

Tracks `ResultMatchEvent` events from the biometric system (LP5 or CBS), extracts candidate information, and sends a JSON to all connected clients via websocket. If no candidate is found, it returns an empty `best_candidate` value. For authentication, it is necessary to add an “Authentication” header with the token from the component information.

The pipeline is necessary for solutions when Access needs to pass data from linked components to external systems beyond the proposed integrations.

The template of the returned data:

```
{
  source: str, best_candidate:
  {
    person_id: str, fullname: str | None, descriptor_id: str | None =
    None
  }
}
```

- source - device name;
- best_candidate - candidate data;

- person_id - candidate ID in ACS;
- fullname - candidate full name;
- descriptor_id - descriptor ID;

To create a pipeline, you need to specify (Table 104):

Table 104. Pipeline settings

Parameter	Description	Possible values	Default value
name	Pipeline name specified by the user	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
vl_access_host	IP address of the server where Access is installed	IP address in the format X.X.X.X or site.domain	-
vl_access_port	Server port where Access is deployed	-	9091

22.9. SendCardToR20Face

The pipeline listens to the queue of SuccessMatchEvent events, validates the received event for the presence of a candidate, the level of compliance, and the presence of a card number. Then the device is searched by the name of the event source and the card number is sent to this device.

To create a pipeline, you need to specify (Table 105):

Table 105. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-

22.10. SendCarsToLaurent

A pipeline for sending events from LunaCars to Laurent. Listens to the Event event queue and generates SigurCarEvent events.

To create a pipeline, you need to specify (Table 106):

Table 106. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
controller_id	UUID of Laurent controller in Access	-	-
relay_{N}_scenario_id	UUID Scenario in LUNA CARS Analytics, according to which the license plate is checked for the relay {N}.	-	-

22.11. SendCarsToSigur

A pipeline for sending events from LunaCars to Sigur. Listens to the Event event queue and generates SigurCarEvent events.

To create a pipeline, you need to specify (Table 107):

Table 107. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-

22.12. SendThermalEventToLuna

Listens to the heat detection event queue and generates events in Luna. To launch, specify the `component_id` of the Luna service running in the system. The pipeline works with several lists: default list and blacklist. It distributes the received data between lists depending on the set thresholds for the lower and upper temperature values.

To create a pipeline, you need to specify (Table 108):

Table 108. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
luna_id	Luna service ID in the system	-	-
handler_id	UUID of the passage event handler, created in Luna	-	-
black_list_id	Luna list ID with which the device will synchronize people who are denied access (optional)	The ID of the list created in Luna	-
to_high_temperature	The upper threshold of a person's temperature at which a person cannot be allowed to pass	3## 37	
to_low_temperature	The lower threshold of a person's temperature at which a person cannot be allowed to pass	3## 35	
use_lists		On — match; Off — do not match	Off
min_face_similarity	Minimum face similarity threshold when performing verification.	The value is formed at the design stage and corrected at the testing stage.	0.00...1.00

22.13. SendToBars

Listens to the LunaEvent, DoorEvent event queues and generates the BarsEvent event.

To create a pipeline, you need to specify (Table 109):

Table 109. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
luna_id	Drop-down list with Luna service IDs in Access	-	-
bars_host	IP address or network name of the PC with installed Bars server	-	-
bars_port	Bars server port		
enable_ssl	Method of data encryption during transmission over the network. Depends on the type of network in the solution.	On - https Off - http	Off
retry_delay_sec	Pause for reattempting the passage	1...10	5
min_face_similarity	Minimum face similarity threshold when performing verification	The value is formed at the design stage and corrected at the testing stage (0,00...1,00)	0,7

22.14. SendToController

Sends a relay opening signal to the device by the name of the event source.

To create a pipeline, you need to specify (Table 110):

Table 110. Pipeline settings

Parameter	Description	Possible values	Default value
name	User-defined pipeline name	Any text names. Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 30 characters.	-

22.15. SendToDevice

Sends a signal to open the relay to the device by the name of the event source and displays the text on the screen.

To create a pipeline, you need to specify (Table 111):

Table 111. Pipeline settings

Parameter	Description	Possible values	Default value
name	User-defined pipeline name	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
successful_pass_message_template	Message upon successful identification. To display the user's name on the terminal screen, upon successful identification, you must use the FULL name variables. The word order in the welcome message can be any.	Any text names and full NAME variables . Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 24 characters.	Welcome
unsuccessful_pass_message	Message in case of unsuccessful identification	Any text names. Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 24 characters.	The person has not been identified

22.16. SendToGrgFaster

It is designed to interact with the MatchPyPhoto pipeline and the GrgFaster device.

To create a pipeline, you need to specify (Table 112):

Table 112. Pipeline settings

Parameter	Description	Possible values	Default value
name	User-defined pipeline name	Any text names. Latin and Cyrillic characters are supported. It is not recommended to enter more than 30 characters.	-
successful_pass_message_template	Message upon successful identification. To display the user's name on the terminal screen, upon successful identification, you must use the NAME variables. The word order in the welcome message can be any.	Any text names and full NAME variables . Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 24 characters.	-
facility_code	A parameter for entering card numbers for their identification by the system. It will be added for each card number before being sent to the controller	-	-

22.17. SendToLuna

The pipeline sends the received FaceDetectionEvent events to Luna.

To create a pipeline, you need to specify (Table 113):

Table 113. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
luna_id	Luna service ID in the system	-	-
enable_fake_events	Submitting faces with failed liveness to view Liveness hack attempts. To activate, the enable_fake_events parameter in the LunaEventListener pipeline must be active	On - send Off - do not send	Off

22.18. SendToParsec

Listens to the queue of LunaEvent and SuccessMatchEvent events in Luna and generates the ParsecEvent event.

To create a pipeline, you need to specify (Table 114):

Table 114. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
luna_id	Drop-down list with Luna service IDs in Access	-	-

Parameter	Description	Values	Default value
parsec_id	Unique ID of the Parsec device used in the integration.Specified in Access	-	-

22.19. SendToSalto

Listens to the ResultMatchEvent event queue from the MatchByPhoto pipeline, in case of successful face validation from the event, sends request to the Salto service to pass through the access point and displays the result of successful/unsuccessful pass on the screen of device.

To create a pipeline, you need to specify (Table 115):

Table 115. Pipeline settings

Parameter	Description	Possible values	Default value
name	Pipeline name set by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters.	-
successful_pass_message_template	Message upon successful identification. To display the user's name on the terminal screen, upon successful identification, you must use the FULL name variables. The word order in the welcome message can be any.	Any text names and full NAME variables . Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 24 characters.	Welcome
unsuccessful_pass_message	Message in case of unsuccessful identification	Any text names. Latin and Cyrillic alphabet input is supported. It is not recommended to enter more than 24 characters.	The person has not been identified

22.20. SendToSigur

Listens to the queue of LunaEvent and SuccessMatchEvent events in Luna and generates the SigurEvent event.

To create a pipeline, you need to specify (Table 116):

Table 116. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
sigur_id	Unique ID of the Sigur device used in the integration. Specified in Access	-	-

22.21. Strazh2FA

If the controller is not in 2FA mode: Listens to the ResultMatchEvent event queue from the MatchByPhoto pipeline, in case of successful face validation in the event, sends a request to the Strazh service to pass through the access point.

If the controller is in 2FA mode, the events that occurred when reading the card, as well as the usual face detection events, are listened to. They are matched by their source. When the first factor arrives with a certain source, the waiting for the second factor is started, the waiting time is set in the second_factor_expiry_time setting in the StrazhController, as soon as it is received, validation of this pair begins.

To create a pipeline, you need to specify (Table 117):

Table 117. Pipeline settings

Parameter	Description	Values	Default value
name	Pipeline name given by the user	Any text names. Only Latin characters are supported. It is not recommended to enter more than 30 characters	-
waiting_time_exceeded_message	Message when the second factor timed out	Any text names. Latin and Cyrillic input is supported. It is not recommended to enter more than 50 characters.	Timeout exceeded
strazh_id	Dropdown list to select Strazh service ID in Access.	-	-