Questions from the manufacturer, based on the results of studying the document ITSpec\_protocol v1

1. Is the following understanding correct:

"ACS FM / FM ACS" is a server of the manufacturer of mail counters, "SDU FM / FM RCS" is a server created by the Russian Post.

**On the contrary - "ASU FM" is a server (system), "SDU FM" is a FM remote control system of a certain vendor - Posmart**

2. Which server implements the asufm-rcs-service interface and which implements the asufm-rcs-manager interface?

We noticed that the post url is used in asufm-rcs-service.yaml and asufm-rcs-manager.yaml: 'https: //[asufm.russianpost.ru](http://asufm.russianpost.ru/" \t "https://mail.qq.com/cgi-bin/_blank) / ...'

**"Asufm-rcs-manager" - "ASU FM" is a  server (system), and "asufm-rcs-service" - "SDU FM" is a remote control system for FM of a certain vendor.**

3. Please detail the purpose of each information in each asufm-rcs-service and asufm-rcs-manager interface. For example: we don't know what `` publicKey '' is and how FM should use it; `` lock / unlock '', described in `` Change FM status '', how the FM should behave; how to use 'rateTables' and more.

**Updated description under contracts (see attached file).**

4. Please provide details on the usage scenarios and methods for using each of the "asufm-rcs-service" and "asufm-rcs-manager" interfaces. If possible, describe them in specific cases.

**Prepared an explanatory note (see attached file).**

1. When we can test the mail server interface, can we provide a test environment or demo?

**[http://40.114.247.228:8080/swagger-ui/](http://40.114.247.228:8080/swagger-ui/" \t "https://mail.qq.com/cgi-bin/_blank) this is stub”**

**Interaction with Remote Control Systems (RCS)**

The interaction between ASU FM and RCS should be based on a bi-directional API:

1.rcs-manager - service on the ASU FM side, implements API for receiving data from RCS

2.rcs-service - service on the RCS side, implements API for receiving data from

Note: ASU FM is a business data system (counterparties, contracts, cash flows, balances, etc.).ASU FM interacts directly with RCS and does not interact with FM.RCS is a system developed by the FM vendor and directly controlling the FM.

**FM lifecycle management**

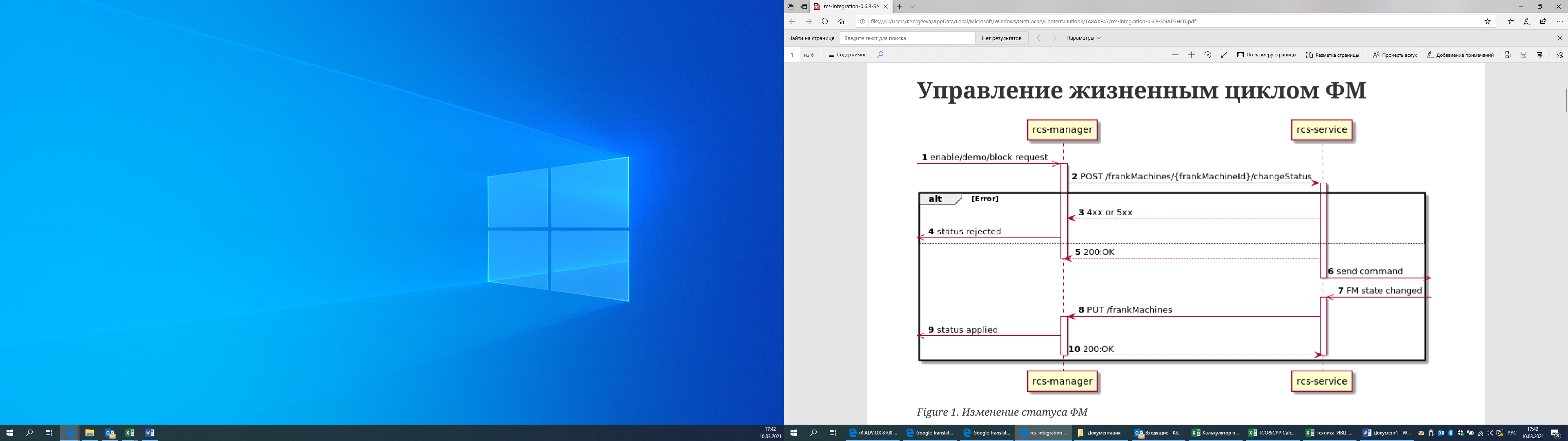


Figure 1. Change of FM status

1. Request for changing the status, initiated by the ASU FM side

2. rcs-manager executes a request for changing the status of a specific FM

3. rcs-service checks the validity of the request and, if the request is not correct, returns an error

4. rcs-manager registers an error and transfers the FM to error state (rejected)

5. If validation of the status change request did not detect an error, rcs-service returns OK (accept)

6. rcs-service in asynchronous mode tries to change the state to FM one

7. When the FM state changes (successful or erroneous), rcs-service receives a notification about this

8. rcs-service updates the FM state on the ACSFM side

9. rcs-manager updates the FM state

10. rcs-manager returns the success of the operation

Before the installation of FM on the PostOffice, the user of the system has already entered the FM card into the system.

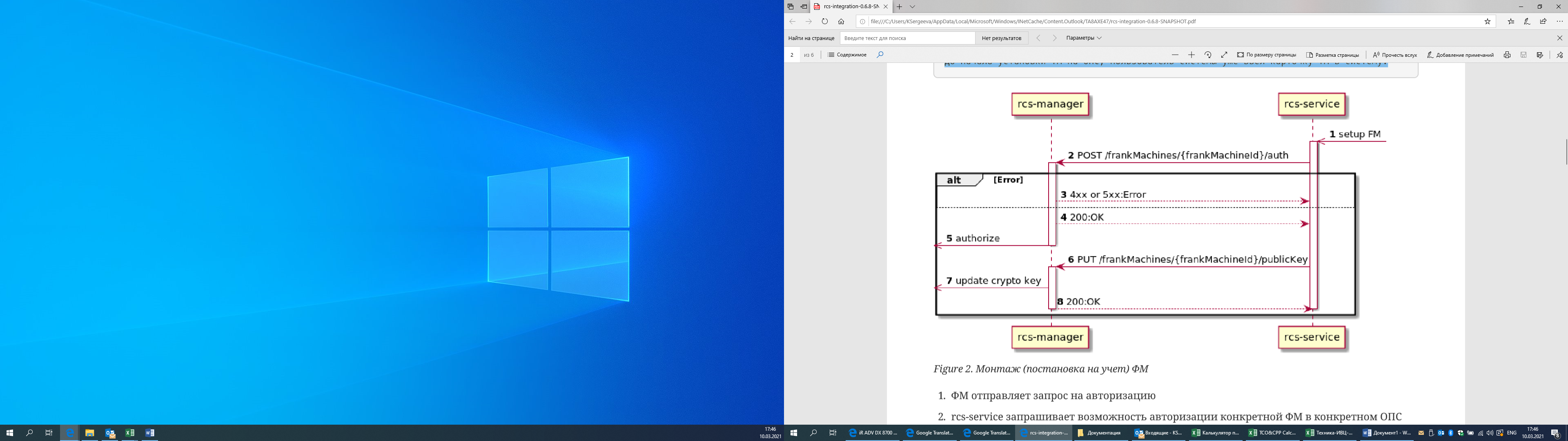


Figure 2. Installation (registration) FM

1. FM sends an authorization request

2.rcs-service requests the ability to authorize a specific FM in a specific post office

3. rcs-manager checks the validity of the request and returns an error if unsuccessful. The error code and text must be available to service personnel at the post office

4.if the request is valid, rcs-manager responds with a 200 code

5. rcs-manager performs authorization in asynchronous mode, which in turn generates the "Change FM status" process described above

6. If the “auth” method is successful, rcs-service publishes the public key in the ACSFM

7. rcs-manager updates the information about the public key

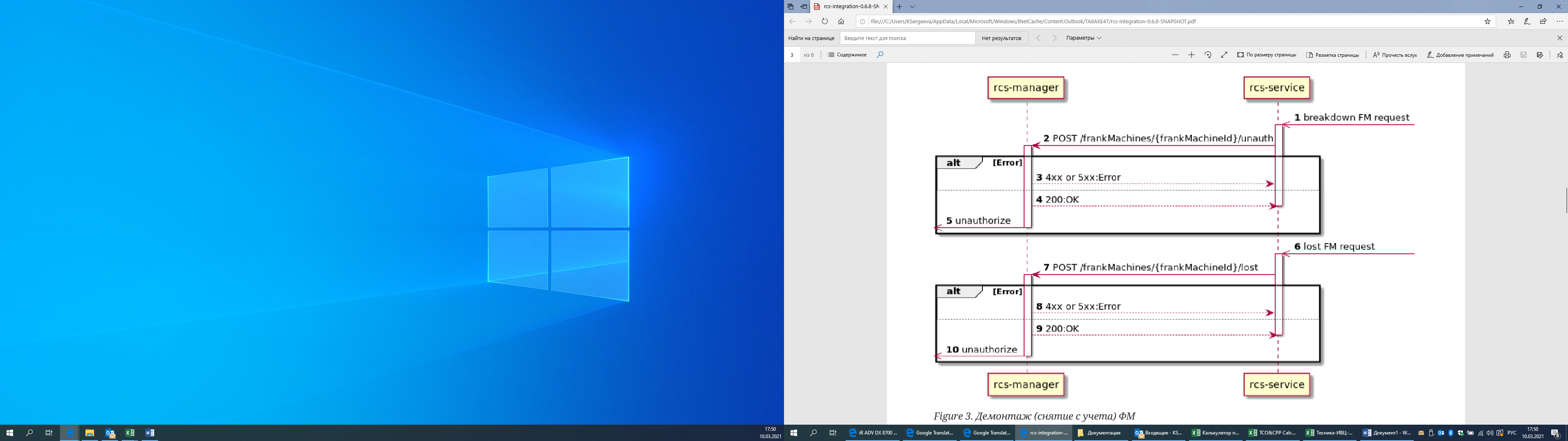


Figure 3. Dismantling (deregistering) FM

1. FM sends a request for deregistration

**2.rcs-service requests the ability to deregister a specific FM in a specific post office**

3. rcs-manager checks the validity of the request data and returns an error if unsuccessful. The error code and text must be available to service personnel at the post office

4.if the request is valid, rcs-manager responds with a 200 code

5. rcs-manager performs deregistration asynchronously. Unlike the registration scenario, the process "Change FM status" is not generated and no additional exchange occurs.

6.A signal is generated about thecompletion of workusing FM

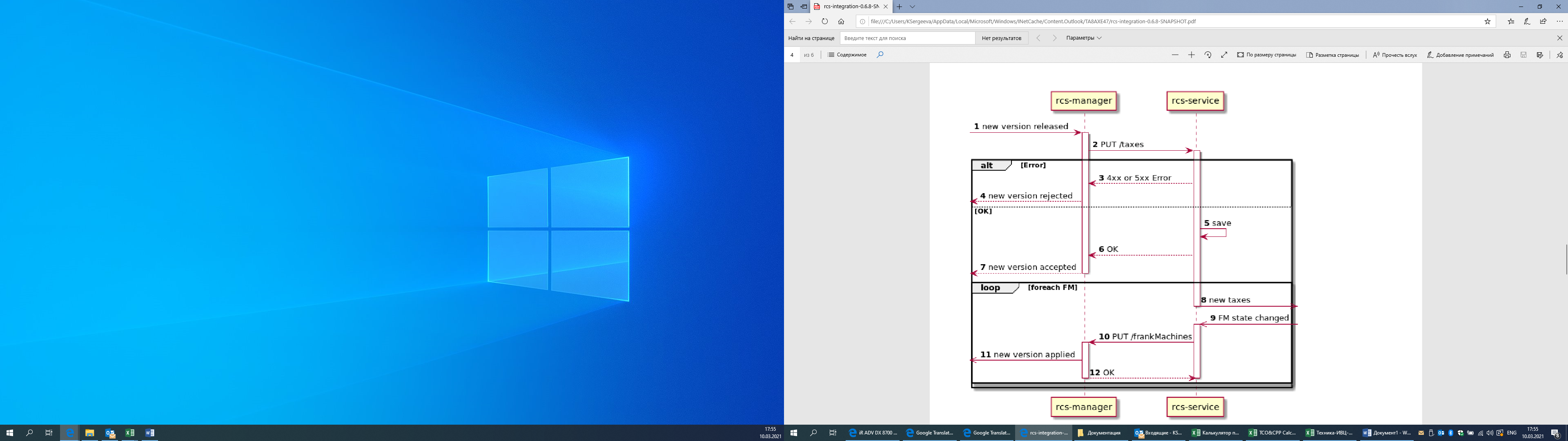
**7.rcs-service requests the ability to** **write off a specific FM**

8. rcs-manager checks the validity of the request data and returns an error if unsuccessful. The error code and text must be available to service personnel at the post office

9.if the request is valid, rcs-manager responds with a 200 code

10. rcs-manager performs the write-off asynchronously. Unlike the registration scenario, the process "Change FM status" is not generated and no additional exchange occurs.

**Tariff (Price List) Management**



1. Central Post Office releases new version of tariffs

2.rcs-manager distributes new tariff table *(price list | taxes)* to SDU (FM Remote Control System) of all vendors *(of Posmart in our case)*

3.rcs-service checks the validity of the request and, if the request is invalid, returns an error

4.rcs-manager logs an error (rejected)

5.rcs-service saves the rate table *(tariff table|price list | taxes)* for further distribution

6.rcs-service responds with status 200

7.rcs-manager logs a positive response

8. **rcs-service for each FM performs the procedure for synchronizing the tariff table**

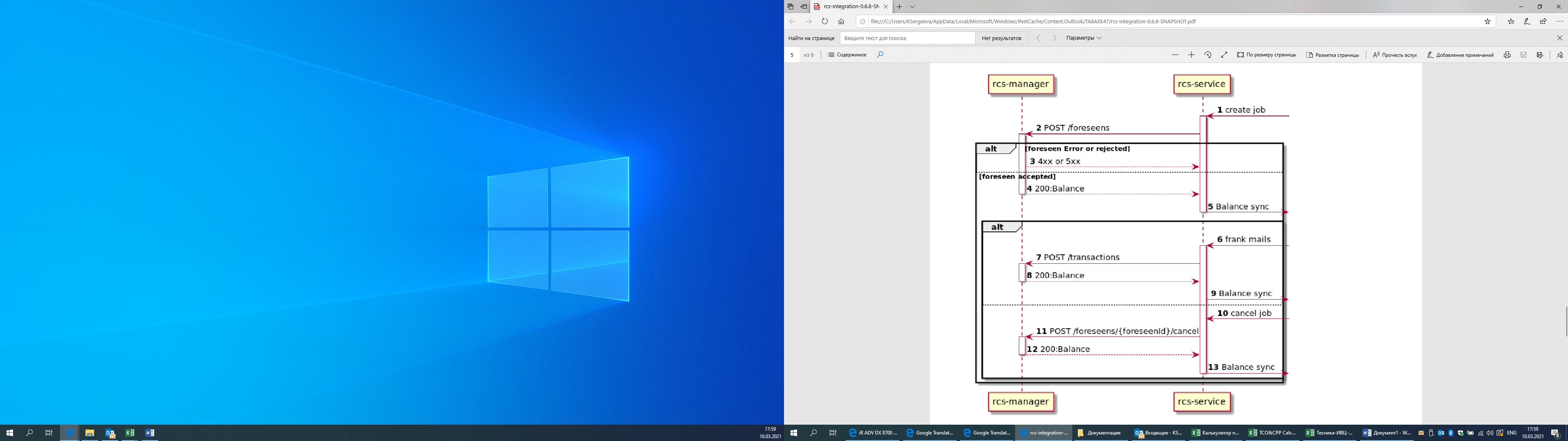
9.What causes changes in the state of FM

10.rcs-service updates the FM state on the ASU FM side

11.rcs-manager updates FM state

12.rcs-manager returns the success of the operation

**Franking process control**



1. The administrator *(FM operator)* creates a task for printing francs

2.rcs-service asks for permission to print

3. rcs-manager validates the request data and responds with an error if unsuccessful. The error text must be available to post office staff

4. rcs-manager gives permission to print the requested francs and reserves the funds in the counterparty's (client’s) account.

5.rcs-service syncs the updated balance

6. The operator performs the franking task according to the pre-approved forecast

7.rcs-service transfers actual work done to rcs-manager

8. rcs-manager confirms data processing. There is a real write-off of funds *(client’s account)*

9.rcs-service synchronizes the updated balance (*of client’s account)*

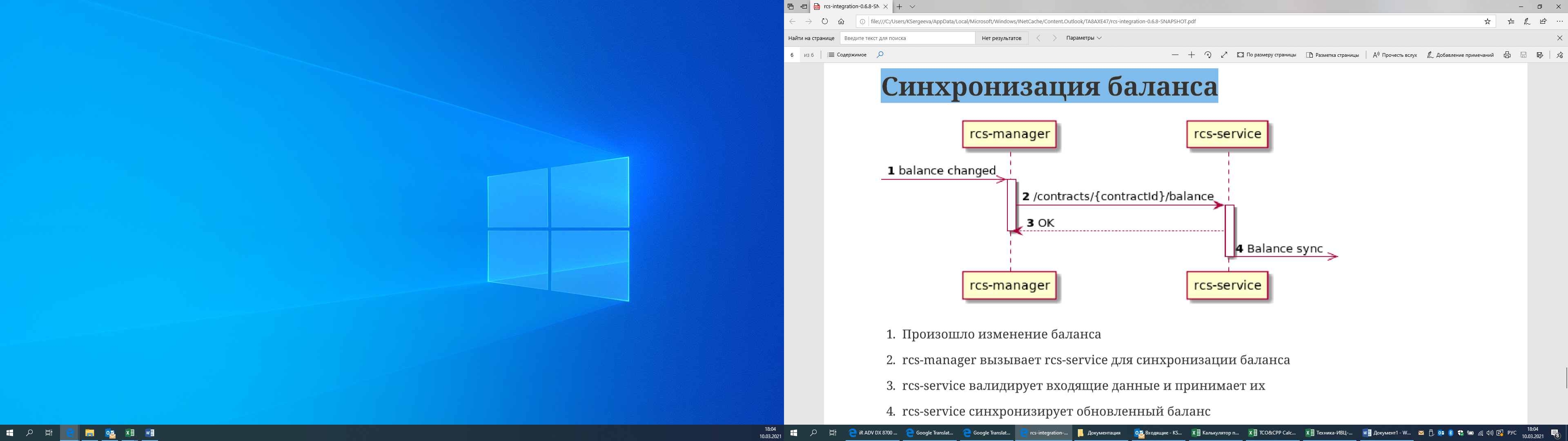
10. The administrator cancels the previously approved franking task.

11.rcs-service passes task cancellation request to rcs-manager

12. rcs-manager confirms data processing. Previously booked funds are canceled

13.rcs-service synchronizes the updated balance

**Balance sync**



1. There was a balance (*of client’s account)* change

2.rcs-manager calls rcs-service to synchronize balance

3.rcs-service validates incoming data and accepts it

4.rcs-service syncs the updated balance

**The different between /rateTables and /frankMachines ("event": "RATE\_TABLE\_UPDATE") method**

The /rateTables method should be initiated on your RCS after successfully uploading new tariffs (TAXES) to the RCS.

The /frankMachines ("event": "RATE\_TABLE\_UPDATE") method should be initiated on the RCS side:

·      after successful authorization of the FM

·      after applying a new tariff (TAXES) on the FM (for example, caused by the date of application of the tariff applyDate (/rateTables))

For add information see attached schemas (fm\_authorisation + taxes\_control).

