

VISIONSPACE

ENVISIONING THE FUTURE

www.visionspace.com

An Open Source Implementation of the Space Link Extension Services in Python

Milenko Starcik

VisionSpace Technologies

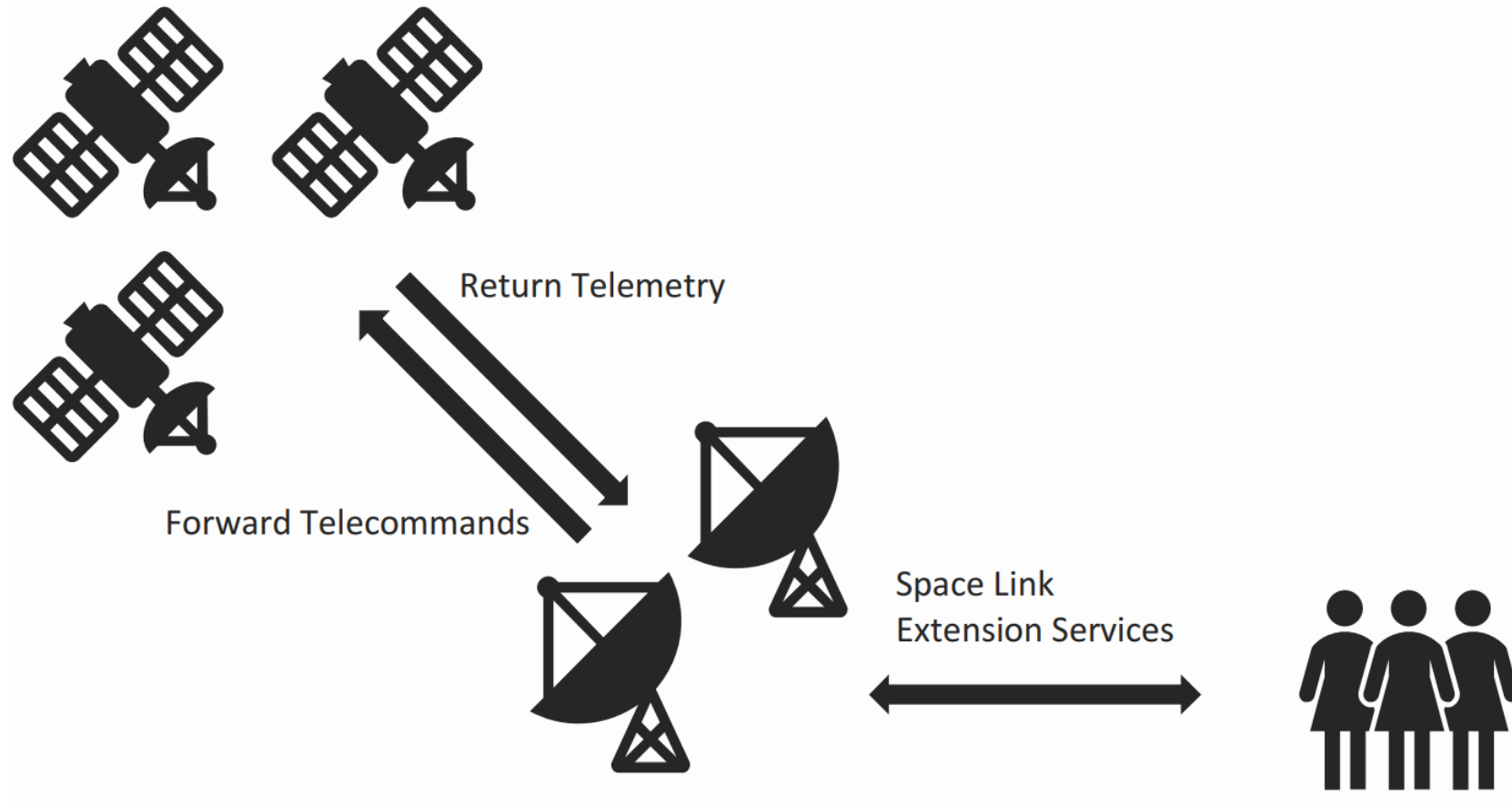
Open Source CubeSat Workshop 2019

Athens, 2019/10/14

Why SLE?

- ✔ No live telemetry for the amateur community
- ✔ Low level of interoperability
- ✔ Only proprietary solutions

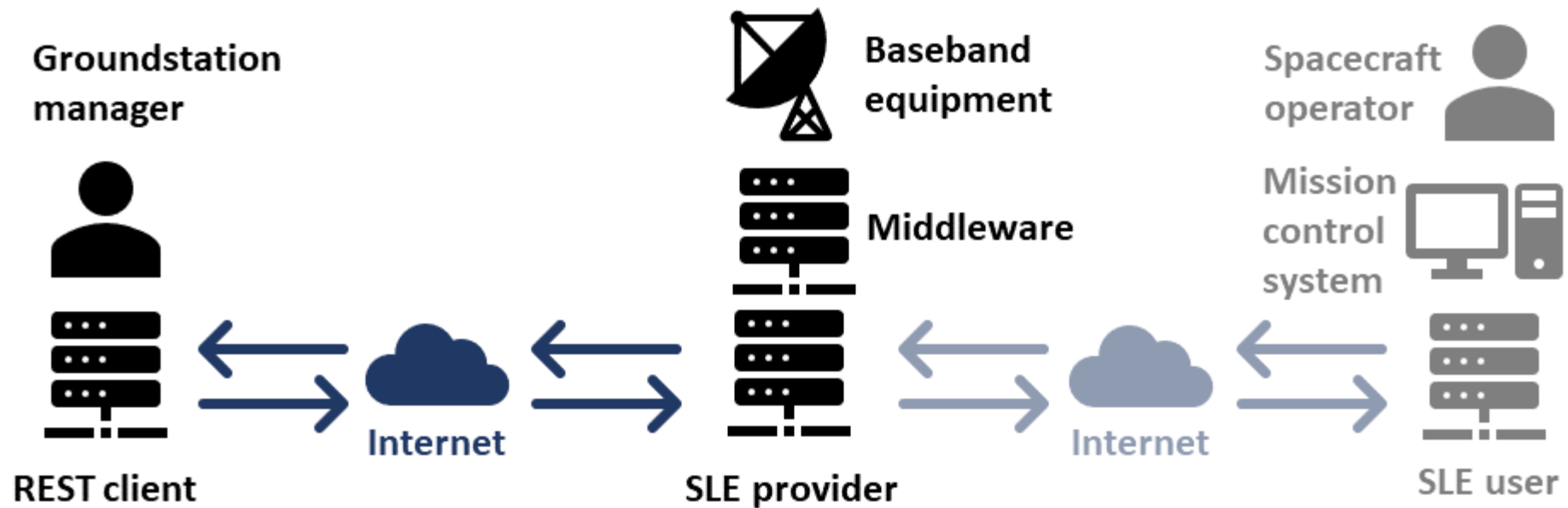
SLE Services



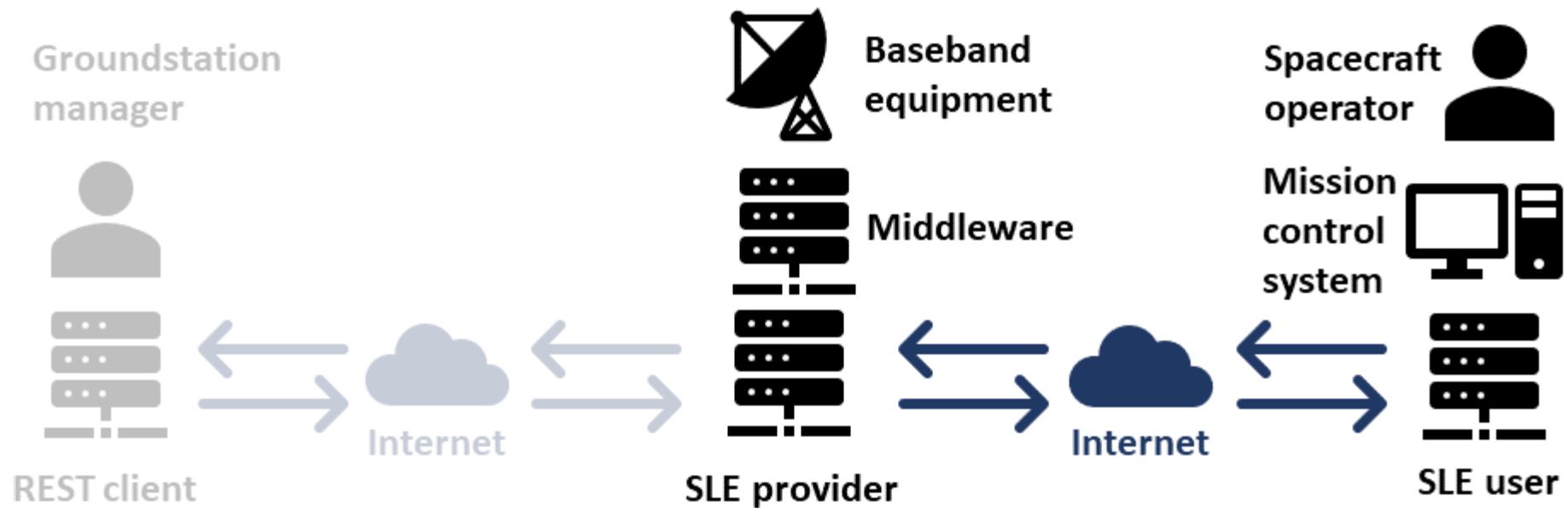
Requirements

- ✔ Free and open source ground systems solution
- ✔ Integration with ground station equipment
- ✔ Convenient configuration

Scenario



Scenario



Start Provider

```
vst@sle-dev: ~/Downloads/sle-provider/examples
File Edit View Search Terminal Help
(venv2) vst@sle-dev:~/Downloads/sle-provider/examples$ python start_provider.py
INFO:sleprovider.sleProvider:rest_manager with http_no_auth_rest_protocol is now running on port: 2048
INFO:sleprovider.sleProvider:sle_provider with sle_protocol is now running on port: 55529
INFO:sleprovider.sleProvider:data_endpoint with json_data_protocol is now running on port: 55555
INFO:sleprovider.sleProvider:SLE Provider is now running!
```


Configure Provider

The screenshot shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a code editor. The code in the editor defines a function to interact with a service instance and a button to trigger the action. Below the code, the rendered UI shows a dropdown menu for 'SI' with the value 'sagr=1.spack=VST-PASS0001' and a 'Show SI' button. The output of the button click is a JSON dictionary containing various configuration parameters.

```
return
print(si_instance.get_si(si_to_read, authentication=auth))

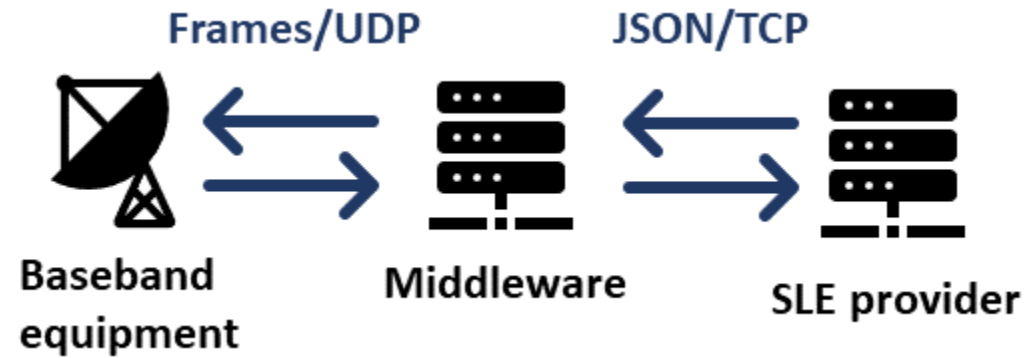
interact(read_select_service_instance, x=widgets Dropdown(options=si_list,
value=next(iter(si_list), None),
description='SI:',
disabled=False,))

display(button_read)
button_read.on_click(read_service_instance)
```

SI:

```
{'delivery_mode': 'TIMELY_ONLINE',
'initiator': 'USER',
'initiator_id': 'SLE_USER',
'latency_limit': 9,
'permitted_frame_quality': ['allFrames', 'erredFramesOnly', 'goodFramesOnly'],
'report_cycle': None,
'requested_frame_quality': 'allFrames',
'responder_id': 'SLE_PROVIDER',
'return_timeout_period': 15,
'start_time': None,
'state': 'unbound',
'stop_time': None,
'transfer_buffer_size': 2}
```

Middleware



- ✓ Frames over UDP
- ✓ JSON based protocol
- ✓ Support for GNU Radio and Cortex

Start Middleware

```
1  from sleprovider.baseband.middleware.gnuRadio import main
2
3  PORT_GOOD_FRAMES = 16887
4  PORT_ERRED_FRAMES = 16888
5  HOST_SLE = 'localhost'
6  PORT_SLE = 55555
7  ANTENNA_ID = 'VST'
8
9  main(PORT_GOOD_FRAMES, PORT_ERRED_FRAMES, HOST_SLE, PORT_SLE, ANTENNA_ID, print_frames=False)
10
```

Start Middleware

```
vst@sle-dev: ~/Downloads/sle-provider/examples
File Edit View Search Terminal Help
(venv2) vst@sle-dev:~/Downloads/sle-provider/examples$ python start_gnuRadio_middleware.py
GNU Radio middleware is now running!
Connection to the SLE provider successful
█
```

SLE Users

- ✔ LibreCube python-sle-user
- ✔ NASA AIT
- ✔ Space Applications Services YAMCS
- ✔ ESA SCOS2000, (EGS-CC)

FOSS
Prop.

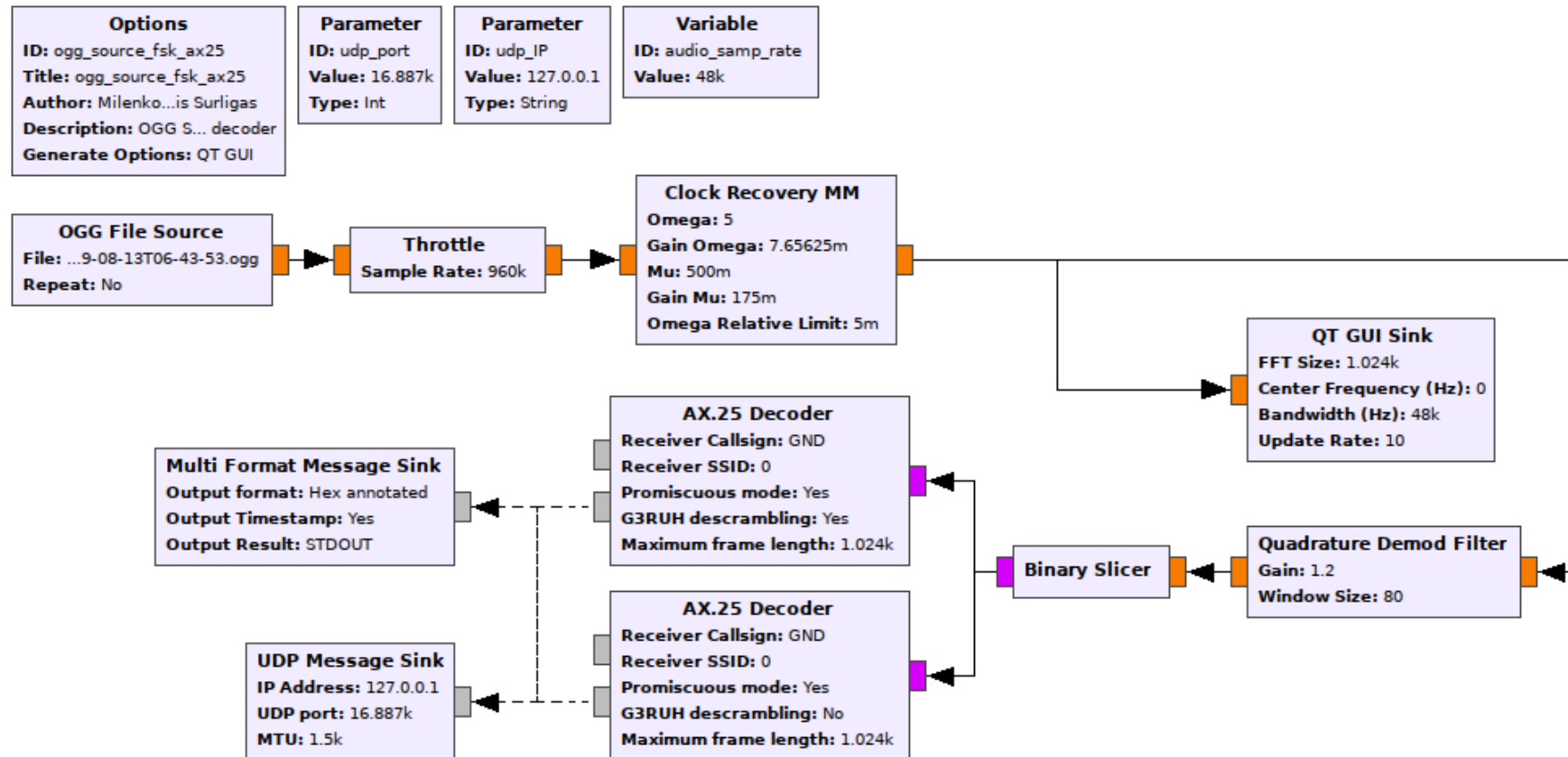
Start User

```
vst@sle-dev: ~/Downloads/python-sle-user/examples
File Edit View Search Terminal Help
(venv) vst@sle-dev:~/Downloads/python-sle-user/examples$ python raf.py
INFO:sle:Sending TML connect request...
INFO:sle:Sending bind request...
INFO:sle:Bind successful
INFO:sle:Sending start invocation...
INFO:sle:Start successful
DEBUG:sle:Heartbeat received
DEBUG:sle:Sending heartbeat
DEBUG:sle:Heartbeat received
DEBUG:sle:Sending heartbeat
```

User Connected

```
vst@sle-dev: ~/Downloads/sle-provider/examples
File Edit View Search Terminal Help
(venv2) vst@sle-dev:~/Downloads/sle-provider/examples$ python start_provider.py
INFO:sleprovider.sleProvider:rest_manager with http_no_auth_rest_protocol is now running on port: 2048
INFO:sleprovider.sleProvider:sle_provider with sle_protocol is now running on port: 55529
INFO:sleprovider.sleProvider:data_endpoint with json_data_protocol is now running on port: 55555
INFO:sleprovider.sleProvider:SLE Provider is now running!
INFO:sleprovider.baseband.dataProtocol:Connection with data endpoint established
INFO:sleprovider.service.commonProtocol:Connection with client established
DEBUG:sleprovider.service.commonProtocol:Context Message received
DEBUG:sleprovider.service.commonProtocol:Bind Invocation received!
DEBUG:sleprovider.service.rafProtocol:Start Invocation received!
DEBUG:sleprovider.service.commonProtocol:Heartbeat received
DEBUG:sleprovider.service.commonProtocol:Heartbeat received
DEBUG:sleprovider.service.commonProtocol:Heartbeat received
DEBUG:sleprovider.service.commonProtocol:Heartbeat received
```

GNU Radio



Live Test

2 - KB9JHU

Owner	Corey Shields
QTH Locator	EM69uf
Coordinates	39.236°, -86.305°
Altitude	280 m
Min Horizon	5°
Target Utilization	100 %
Antennas	<input type="radio"/> UHF Yagi <input type="radio"/> VHF Yagi
Success Rate	<div style="width: 100%;"><div style="width: 100%;"></div></div>
Observations	20296 View all
Creation Date	4 years, 2 months ago
Client version	0.9
Testing	Last seen 0 minutes ago
Uptime	554 days, 3:38:50 Log



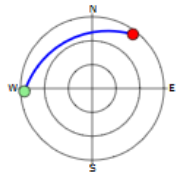
Yaesu G-5500 with M2 cross yagi antennas and S-band parabolic dish

Live Test

Observation #1073861

🕒 Timeframes are in UTC

Discuss

Satellite	42726 - NSIGHT-1
Station	2 - KB9JHU
Observer	Corey Shields
Status	Good
Transmitter	Telemetry
Frequency	435.900 MHz
Encoding	FSK9k6
Timeframe	2019-10-05 15:47:01 2019-10-05 15:53:50
Rise	268.0°
Max	23.0°
Set	37.0°
Client Version	0.9
Metadata	{ 5 items }
Polar Plot	

Downloads

📎 Audio 📎 Waterfall

Waterfall Audio Data 1

ASCII HEX

data_obs/1073861/data_1073861_2019-10-05T15-50-04

```
B4 A6 62 A6 86 A6 E0 9E 9C 60 64 82 B4 61 03 F0 FF 70 06 A8 25 18 00 40 00 00 00 93 C5
03 00 9C 2E BC 28 19 00 00 60 03 00 2E 00 01 00 15 14 14 13 01 03 6D 20 FE 00 85 00 00
00 00 00 00 00 02 00 00 FF 66 CC 09 03 00 40 24 00 00 00 00 00 00 00 33 FE 60 83 17 FA
84 F4 CD C0 45 89 00 00 4F 7B 00 00 00 00 00 00 00 00 00 A8 1E 1C BA 9A 20
```

Results

- ✔ Works with CubeSats and commercial satellites
- ✔ Amateur and professional equipment
- ✔ +1Mbps telemetry throughput

Outlook

- ✔ Performance and security optimization
- ✔ Telecommand uplink
- ✔ Integration with more SatNOGS stations

Thank you

Milenko Starcik
(milenko.starcik@visionspace.com)

VisionSpace Technologies GmbH

Robert-Bosch-Strasse 7

64293 Darmstadt

Germany



twitter.com/visionspacetech

VISION
SPACE

Try it out!



<https://github.com/visionspacetec/sle-provider>