

LibreCube Activities

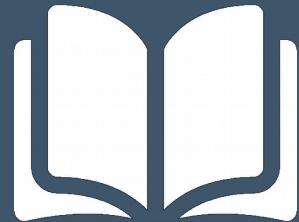
Review and Outlook

Artur Scholz
LibreCube Initiative

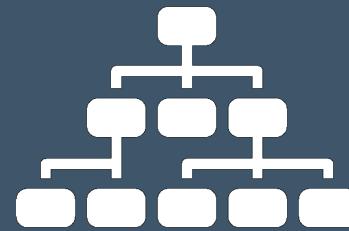
Open Source Space and Earth Exploration



Open Source
Everything

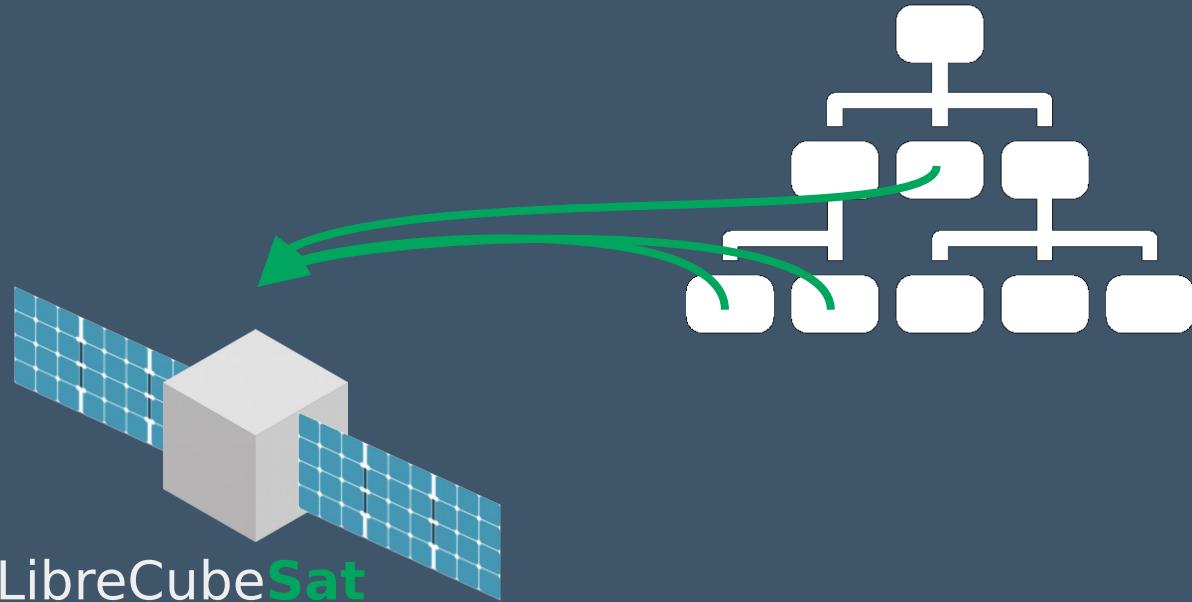


Space
Standards

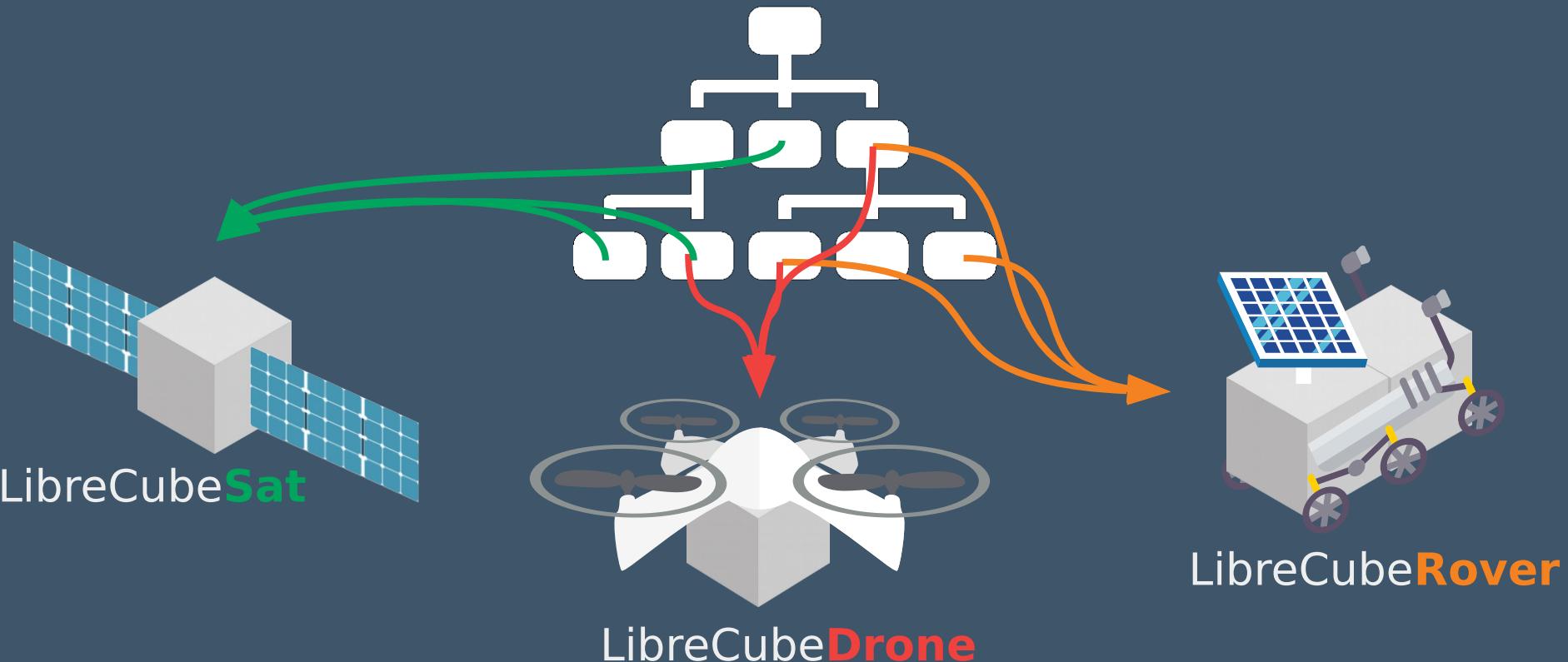


Reference
Architecture

Build your Mission



Build your Mission

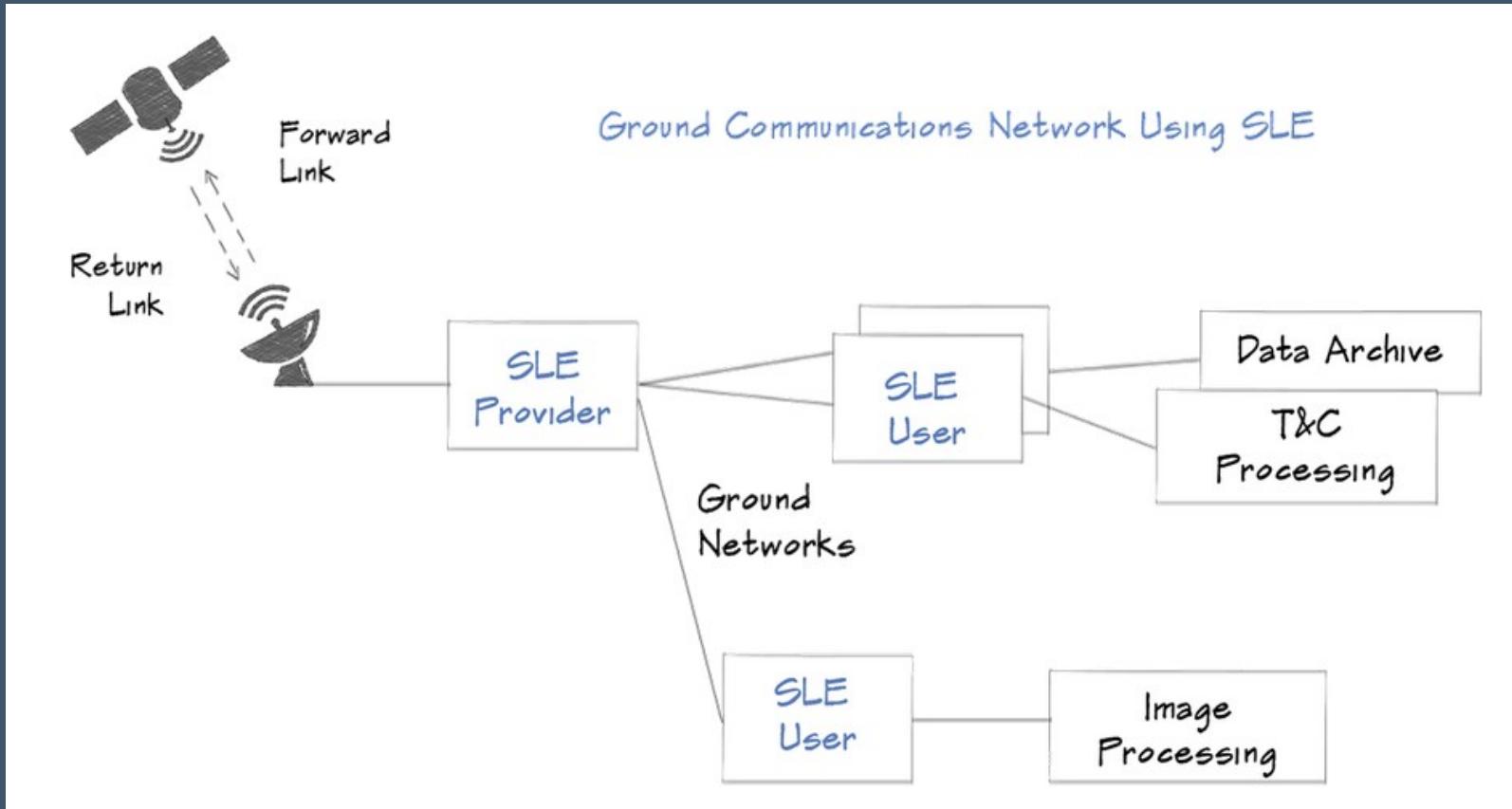


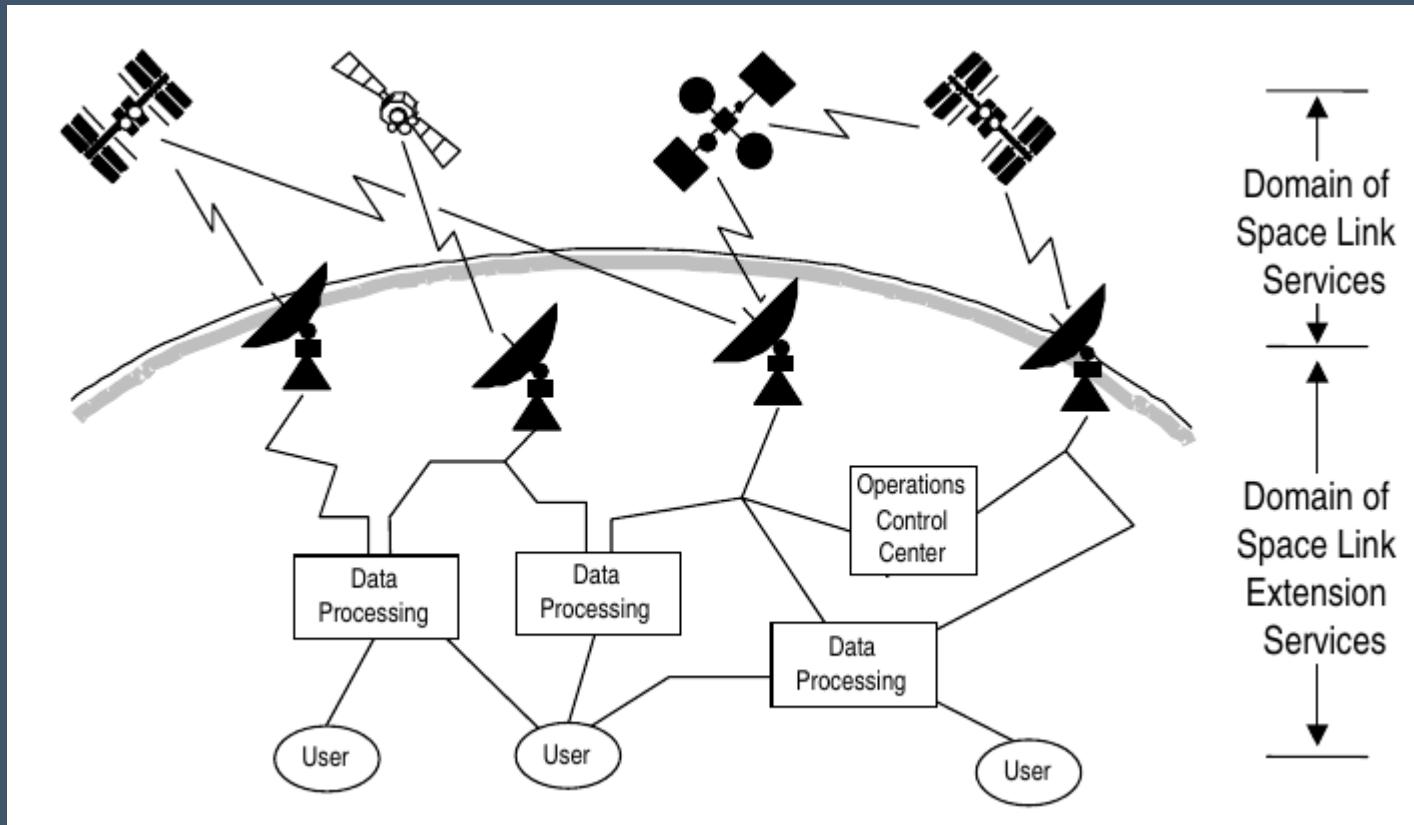
Stuff done

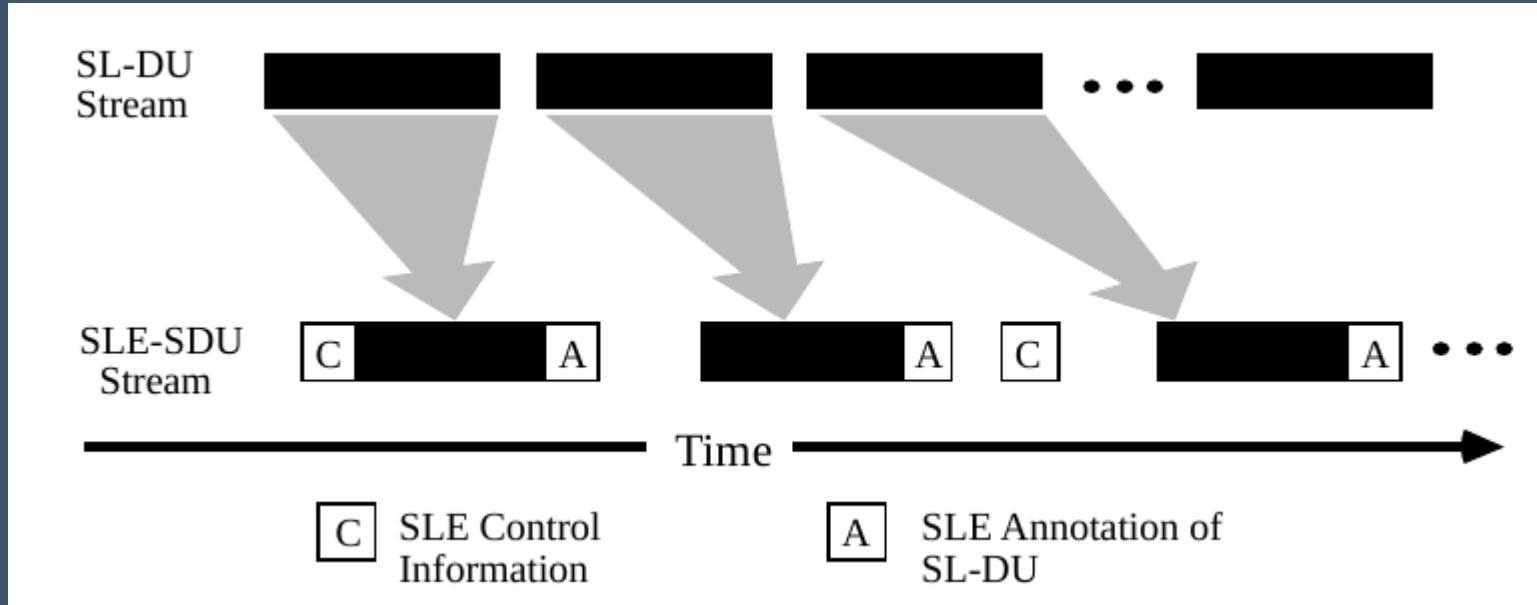
SLE User

Space Link Extension User in Python

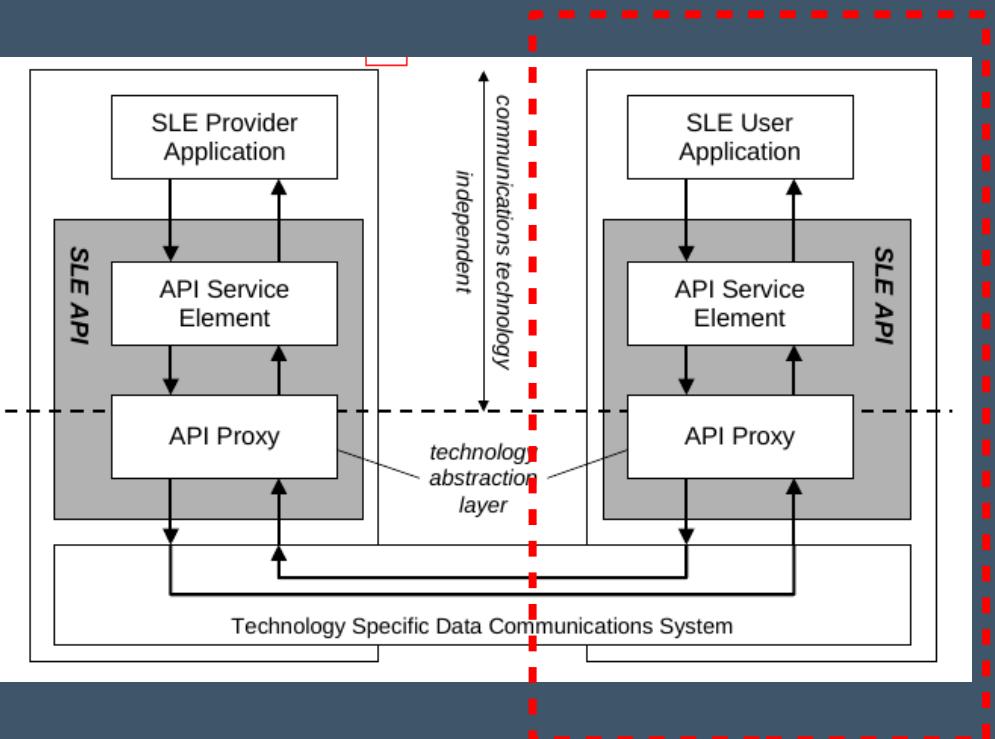
<https://gitlab.com/librecube/lib/python-sle-user>
Milenko Starcik, Fabian Burger, Artur Scholz











```

import sle

raf = sle.RafUser(
    service_instance_identifier=os.environ.get('RAF_INST_ID'),
    responder_ip=os.environ.get('SLE_PROVIDER_HOSTNAME'),
    responder_port=int(os.environ.get('SLE_PROVIDER_TM_PORT')),
    auth_level='bind',
    local_identifier=os.environ.get('INITIATOR_ID'),
    peer_identifier=os.environ.get('RESPONDER_ID'),
    local_password=os.environ.get('PASSWORD'),
    peer_password=os.environ.get('PEER_PASSWORD')
)

raf.bind()
raf.start()
time.sleep(5)
raf.stop()
raf.unbind(reason='other')

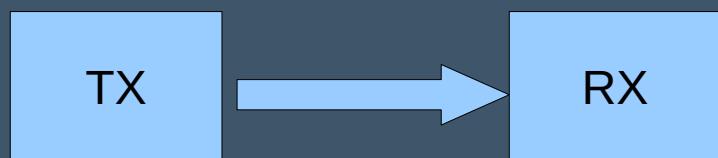
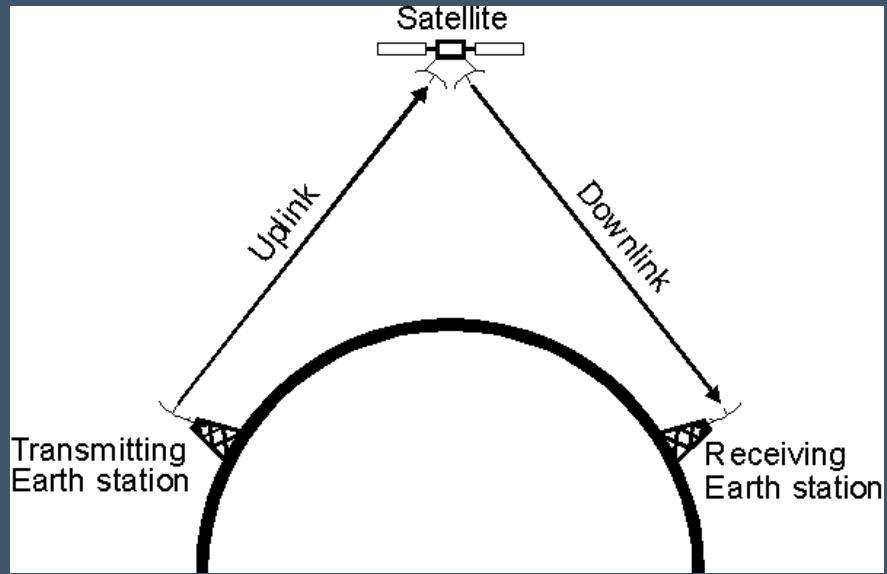
```

Linkpredict

Generic and Modular Link Budget in Python

<https://gitlab.com/librecube/lib/linkpredict>
Jona Saffer, Artur Scholz

Linkpredict



```
import linkpredict as lp

channel = lp.Channel(...)
# ...
link = lp.Link(
    channel=channel,
    geometry=geometry,
    transmitter=transmitter,
    transmit_antenna=transmit_antenna,
    receive_antenna=receive_antenna,
    receive_antenna_noise=receive_antenna_noise,
    receiver=receiver,
    medium_losses=medium_losses,
)
result = link.calculate_link_budget()
```

Linkpredict

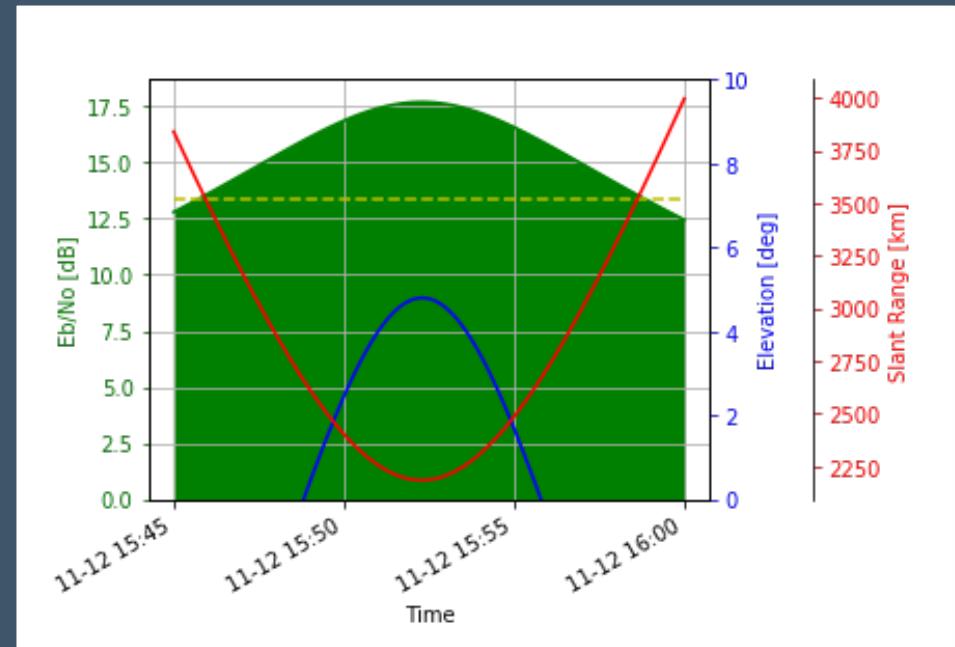
```
# Transmitter (on satellite)
onboard_losses = lp.Device(gain=-1)
amplifier_power = 0 # 0 dBW = 1 Watt
transmitter = lp.Transmitter(amplifier_power, [onboard_losses])
tx_antenna = lp.OmniDirectionalAntenna(gain=0, linear_polarized=True)

# Geometry
spacecraft = lp.SpacecraftObject()
name = "CUBEBEL-1 (BSUSAT-1)"
line1 = "1 43666U 18083E 18314.15998747 .00001095 00000-0 58587-4 0 9994"
line2 = "2 43666 97.5398 334.9753 0013890 232.5130 215.5169 15.17110642 1849"
spacecraft.set_orbit_from_tle([name, line1, line2])
groundstation = lp.GroundstationObject()
lat, lon, alt = 50.750, 6.216, 275
groundstation.set_location(lat, lon, alt)
geometry = lp.GroundstationSpacecraftGeometry(groundstation, spacecraft)

# Path
atmospheric_loss = lp.SimpleMediumLoss(1)
medium_losses = [atmospheric_loss]

# Channel
modulation = lp.FSKNonCoherentNoCoding(bit_rate=9600)
channel = lp.Channel(436.2e6, modulation=modulation)

# Receiver (groundstation)
line_losses = lp.Device(gain=-2.0)
receiver = lp.Receiver.from_noise_figure(noise_figure=2.0, devices=[line_losses])
rx_antenna = lp.MainLobeAntenna(peak_gain=15.5, beam_3db_width=5)
rx_antenna_noise = lp.SimpleAntennaNoise(300)
```



PLUTO to Python

A Parser of ECSS PLUTO scripts

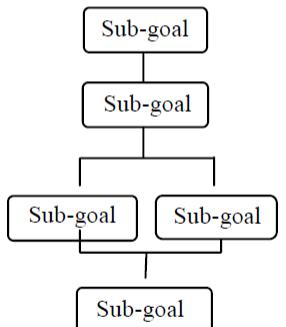
<https://gitlab.com/librecube/prototypes/python-pluto>
Vudit Jain, Christoph Buchner, Artur Scholz

PLUTO to Python Parser

Declaration Body

Preconditions Body

Main Body
Sequential sub-goals



Parallel sub-goals

Watchdog Body



Confirmation Body

procedure

preconditions

 wait until value of GyroTemp > 60 degC

end preconditions

main

 initiate and confirm SwitchOnGyro;

end main

confirmation

 wait until value of Gyro5 < 0.2 deg/h

end confirmation

end procedure

PLUTO to Python Parser

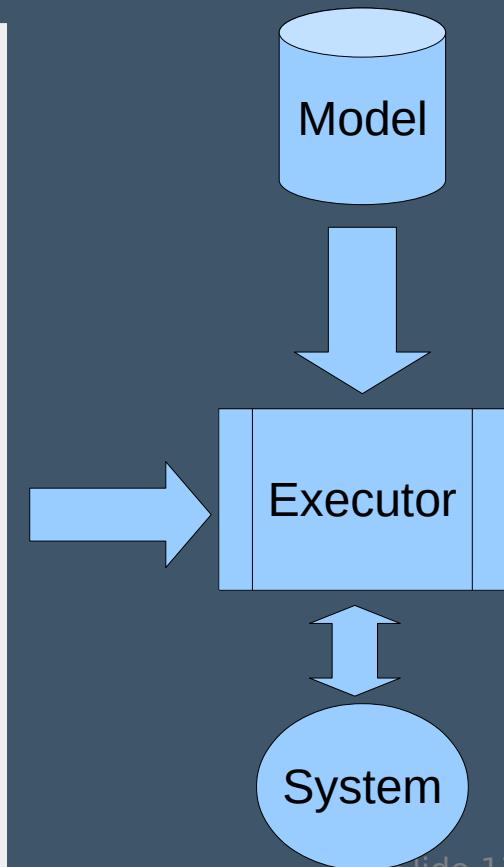
```
from pluto import *
from model import *

class Procedure_test0921(Procedure):

    def preconditions(self):
        if self.wait_until_expression(
            lambda: GyroTemp.get_value() > ureg('60degC')) is False:
            return False

    def main(self):
        act = ActivityCall(self, SwitchOnGyro)
        if self.initiate_and_confirm_activity(act) is False:
            return False

    def confirmation(self):
        if self.wait_until_expression(
            lambda: Gyro5.get_value() < ureg('0.2deg')/h) is False:
            return False
```



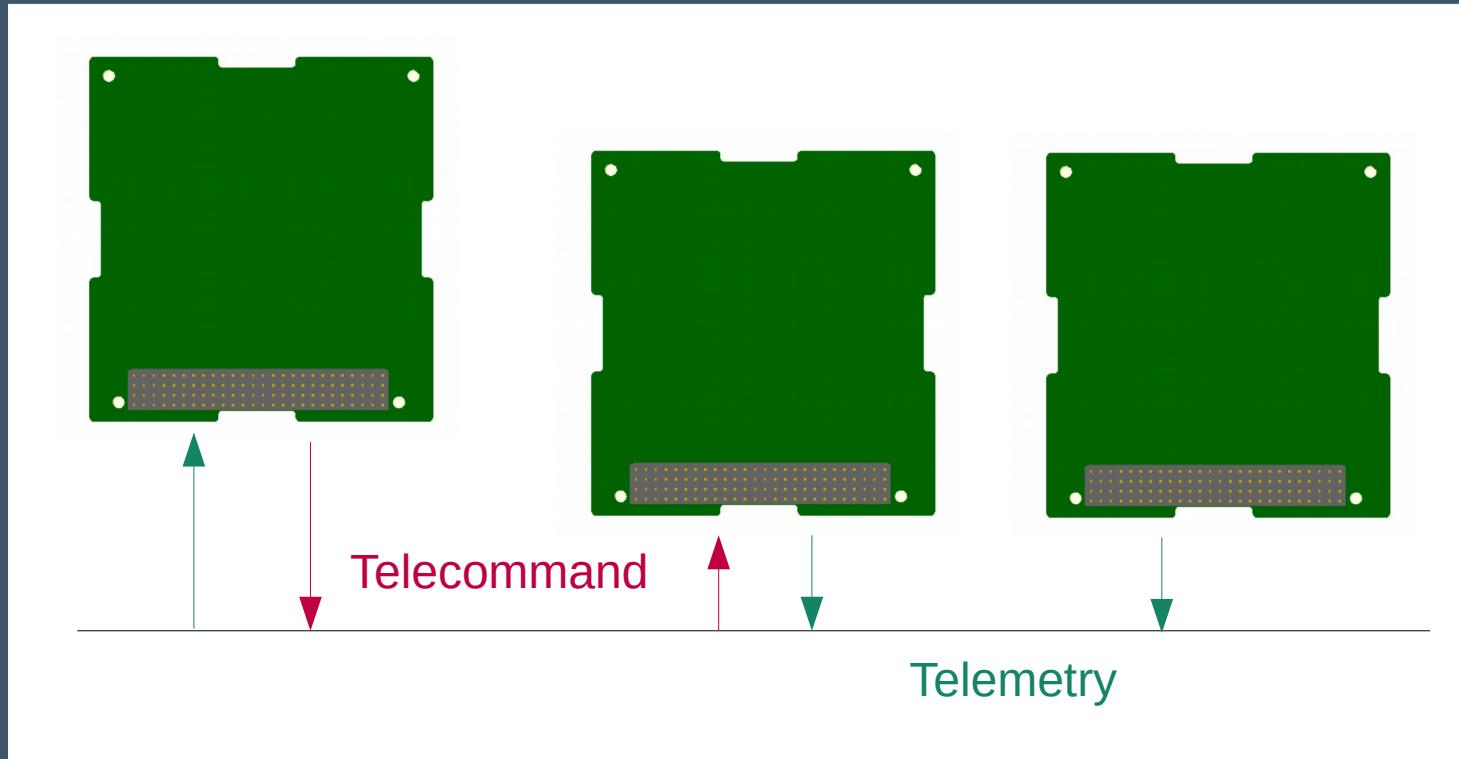
SpaceCAN

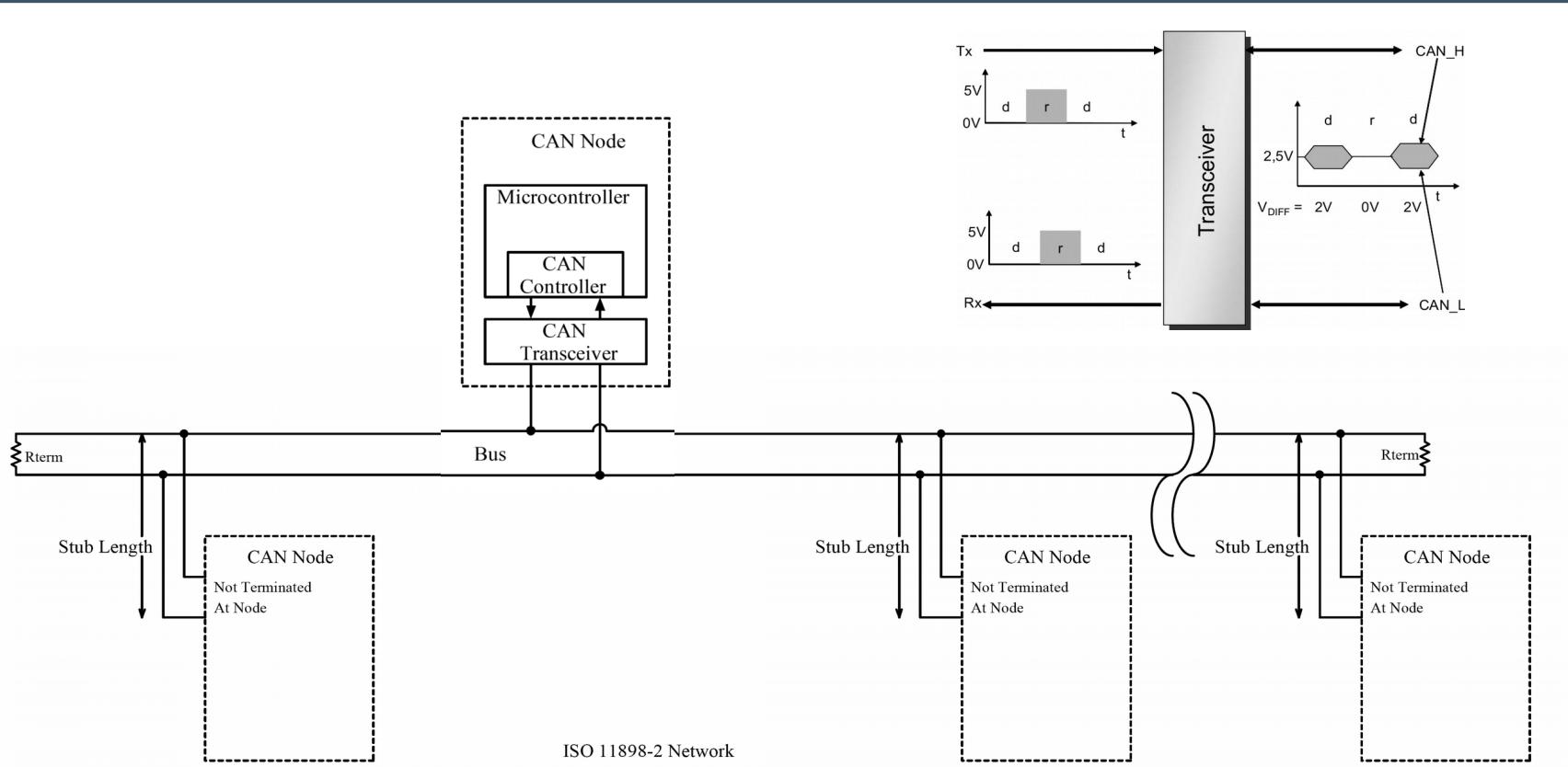
Robust and Reliable System Bus

<https://gitlab.com/librecube/lib/micropython-spacecan>

<https://gitlab.com/librecube/lib/cpp-spacecan>

Peter Mader, Jesper Schlegel, Nicolas Wolf, Artur Scholz





Services

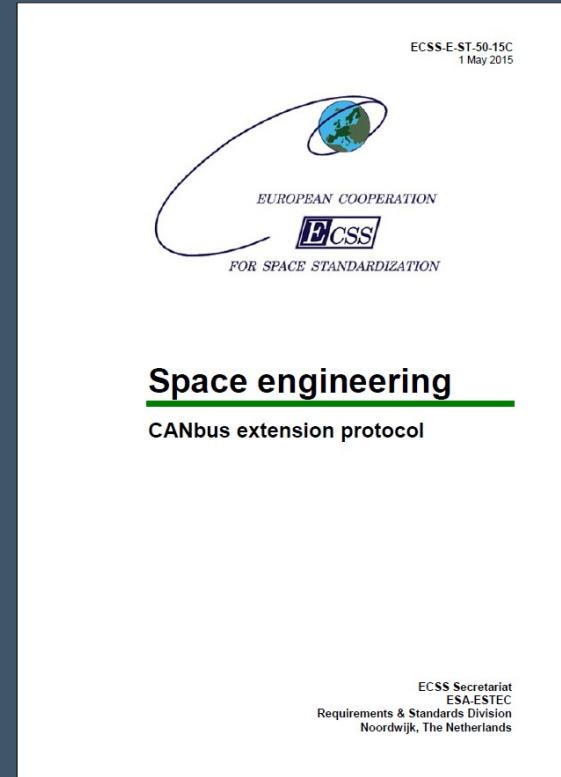
Redundancy management

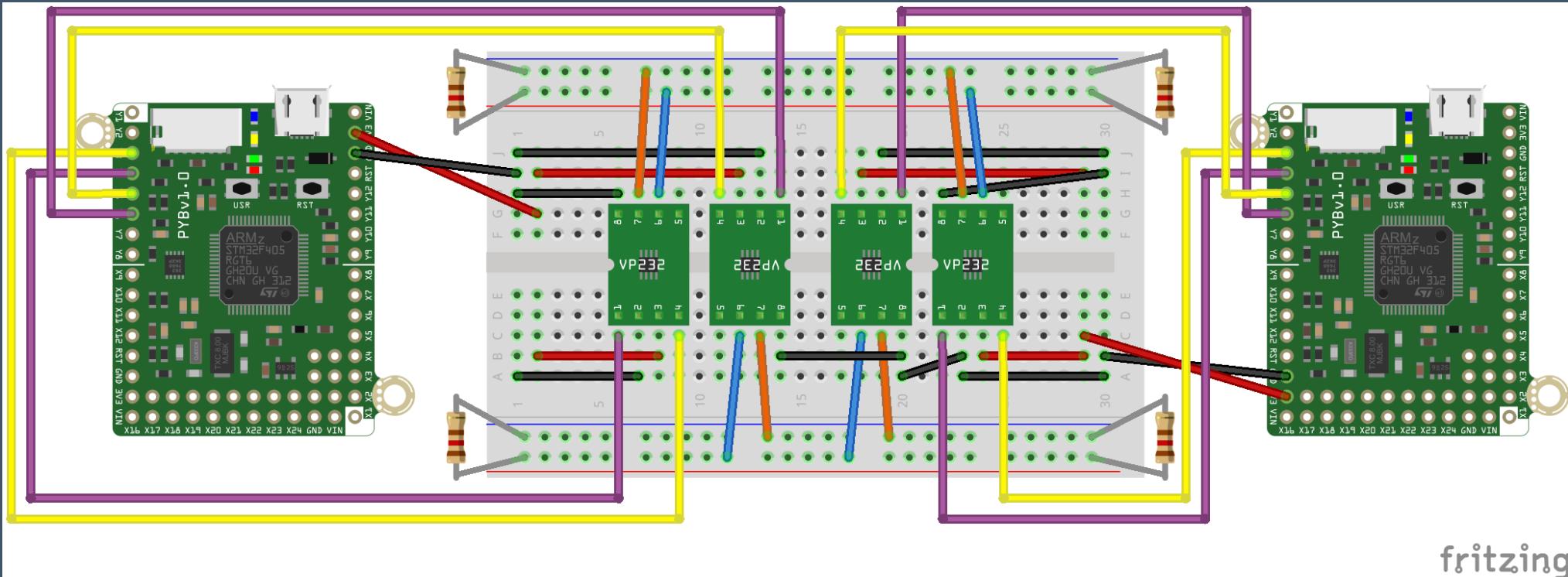
Synchronization

Time distribution

TC/TM exchange

(Message exchange)



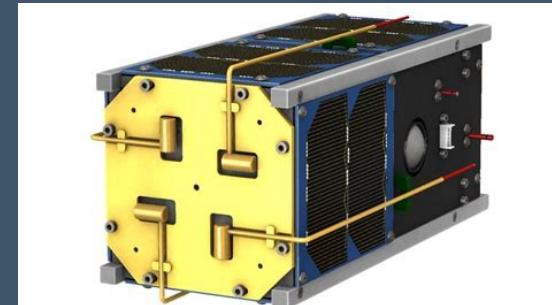
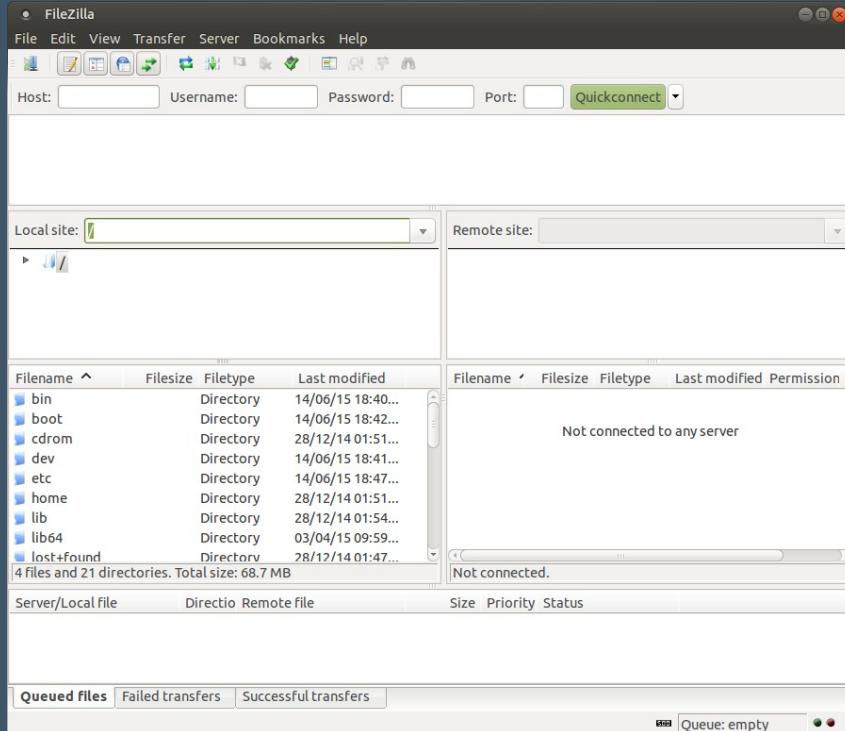


Stuff ahead

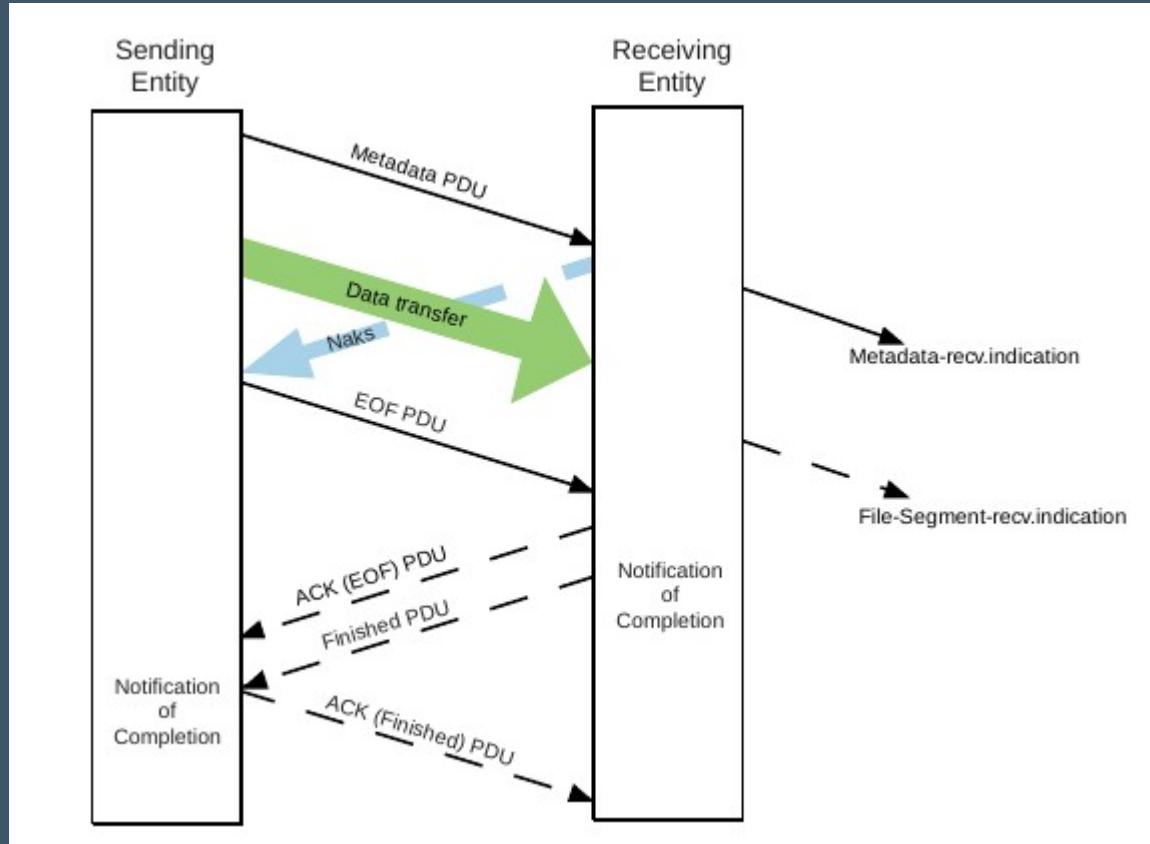
CDFP in Python

CCSDS File Delivery Protocol

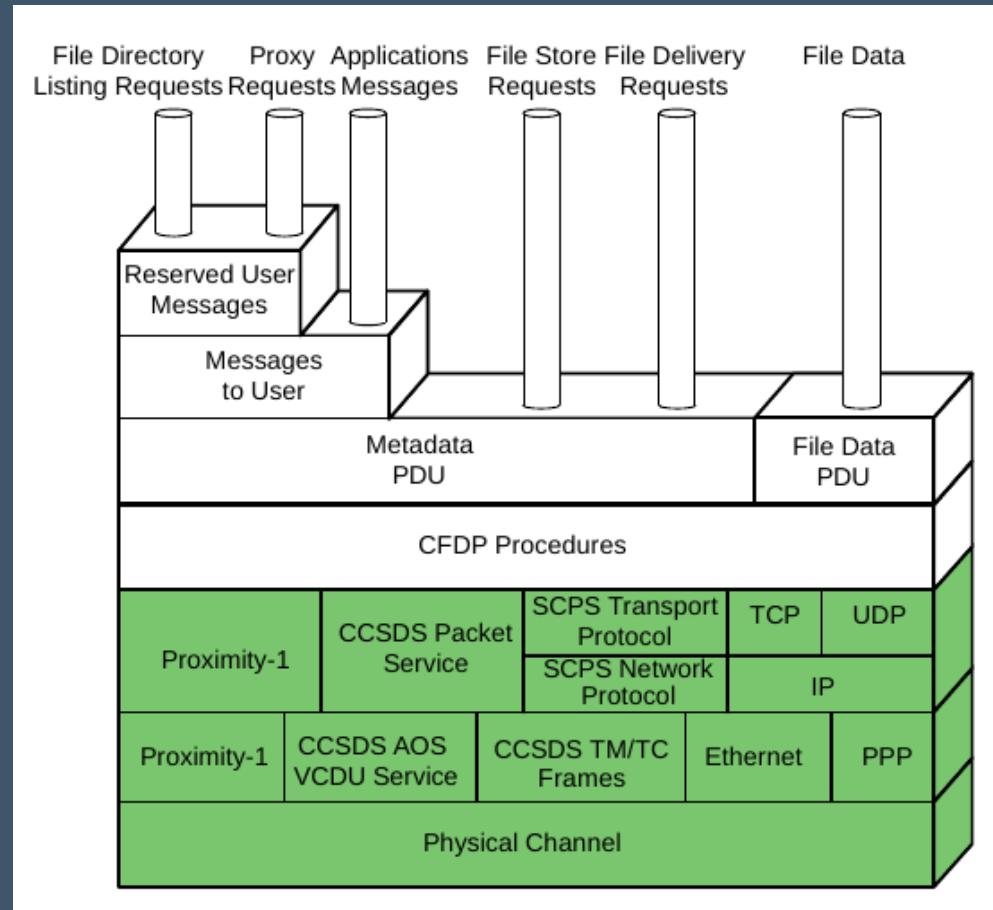
CCSDS File Delivery Protocol



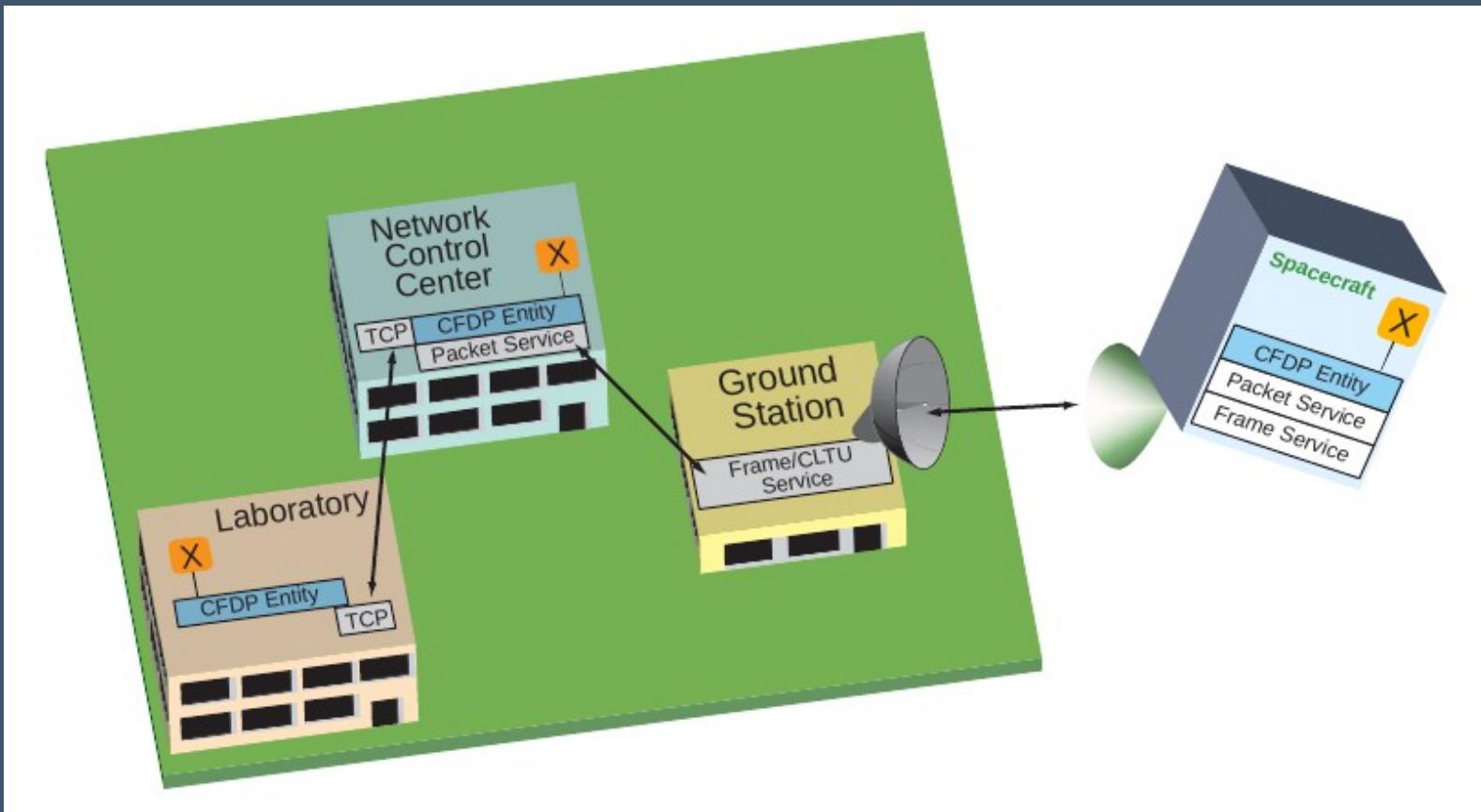
CCSDS File Delivery Protocol



CCSDS File Delivery Protocol



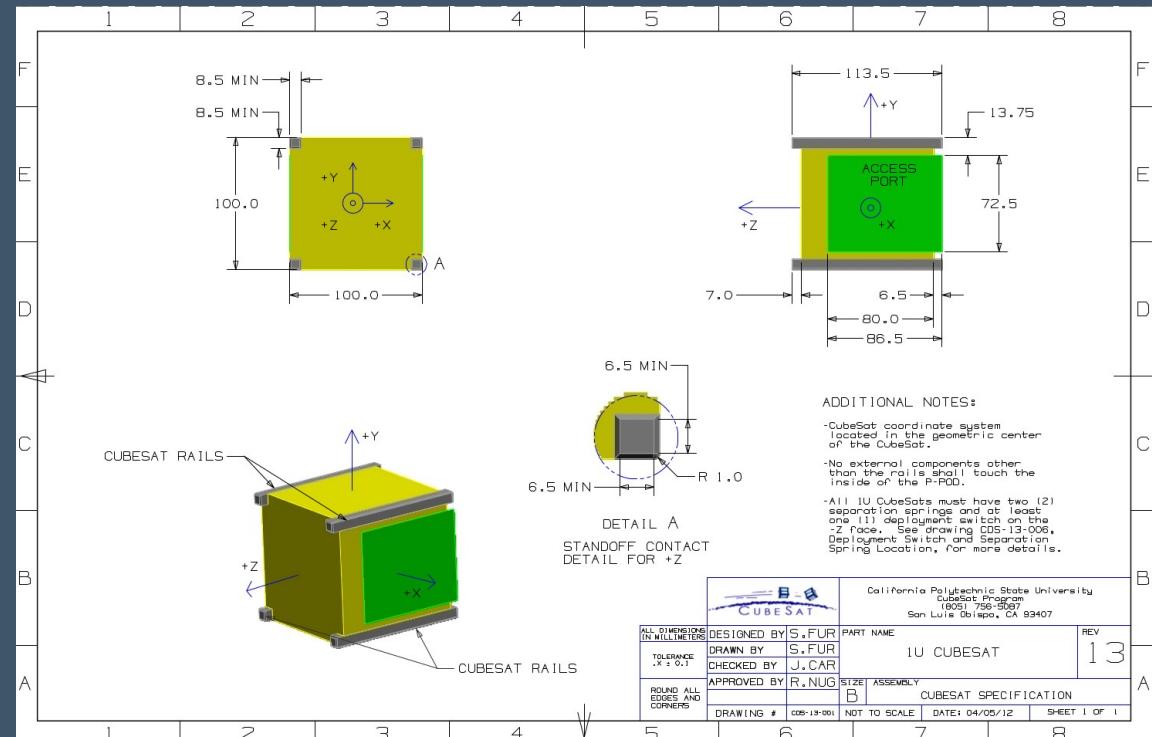
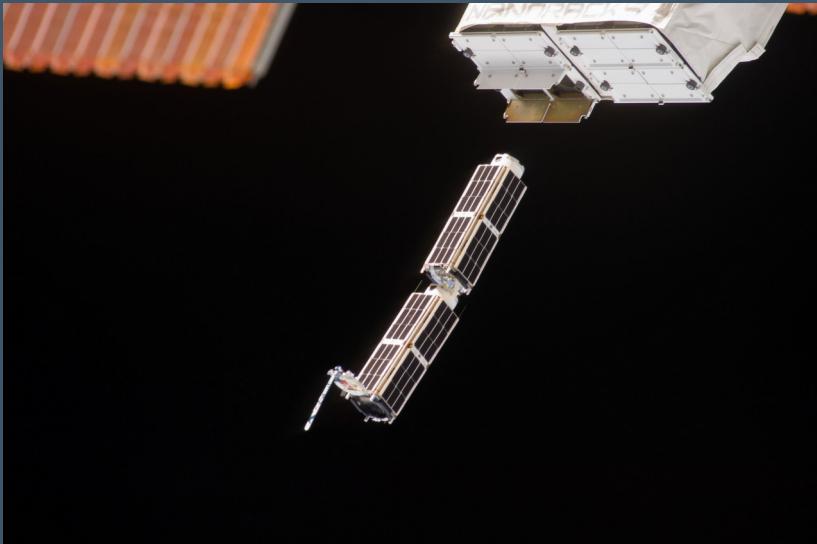
CCSDS File Delivery Protocol



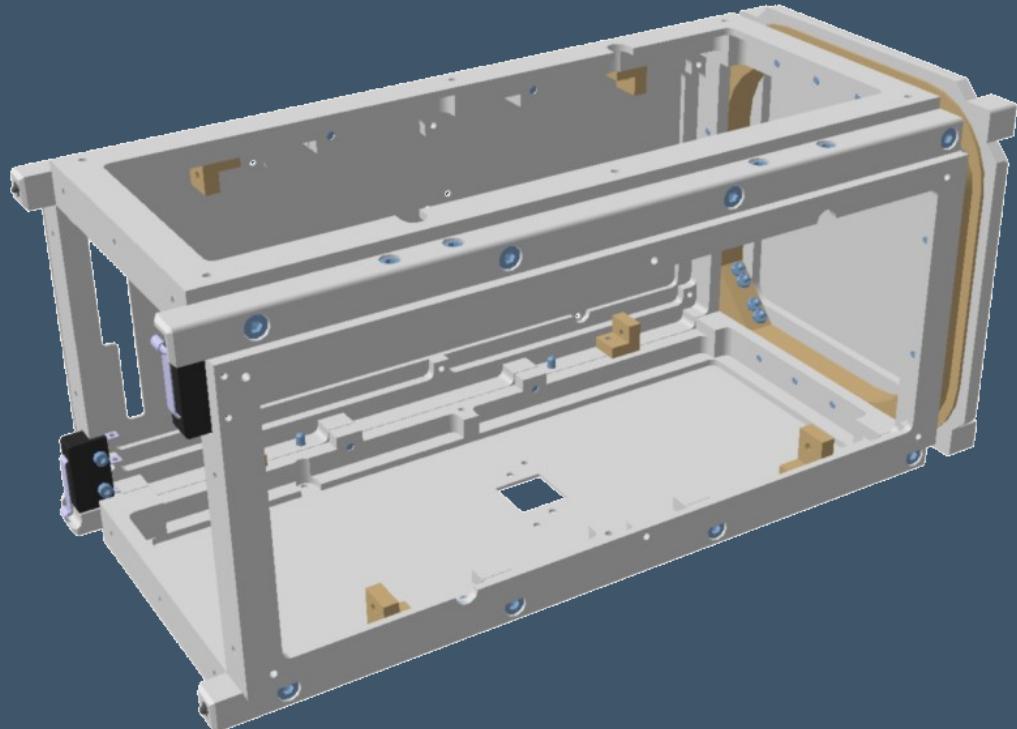
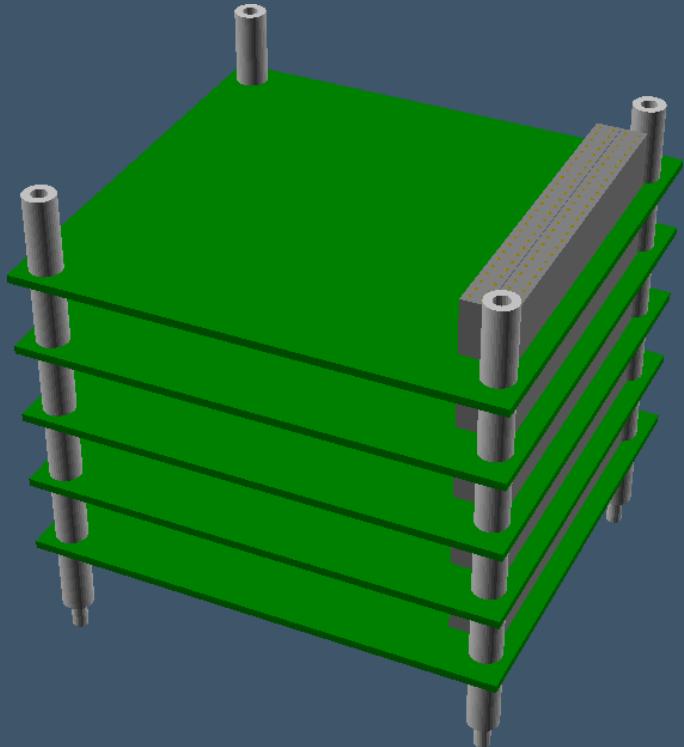
1U Structure

Modular, Extensible 1U CubeSat Structure

1U Structure



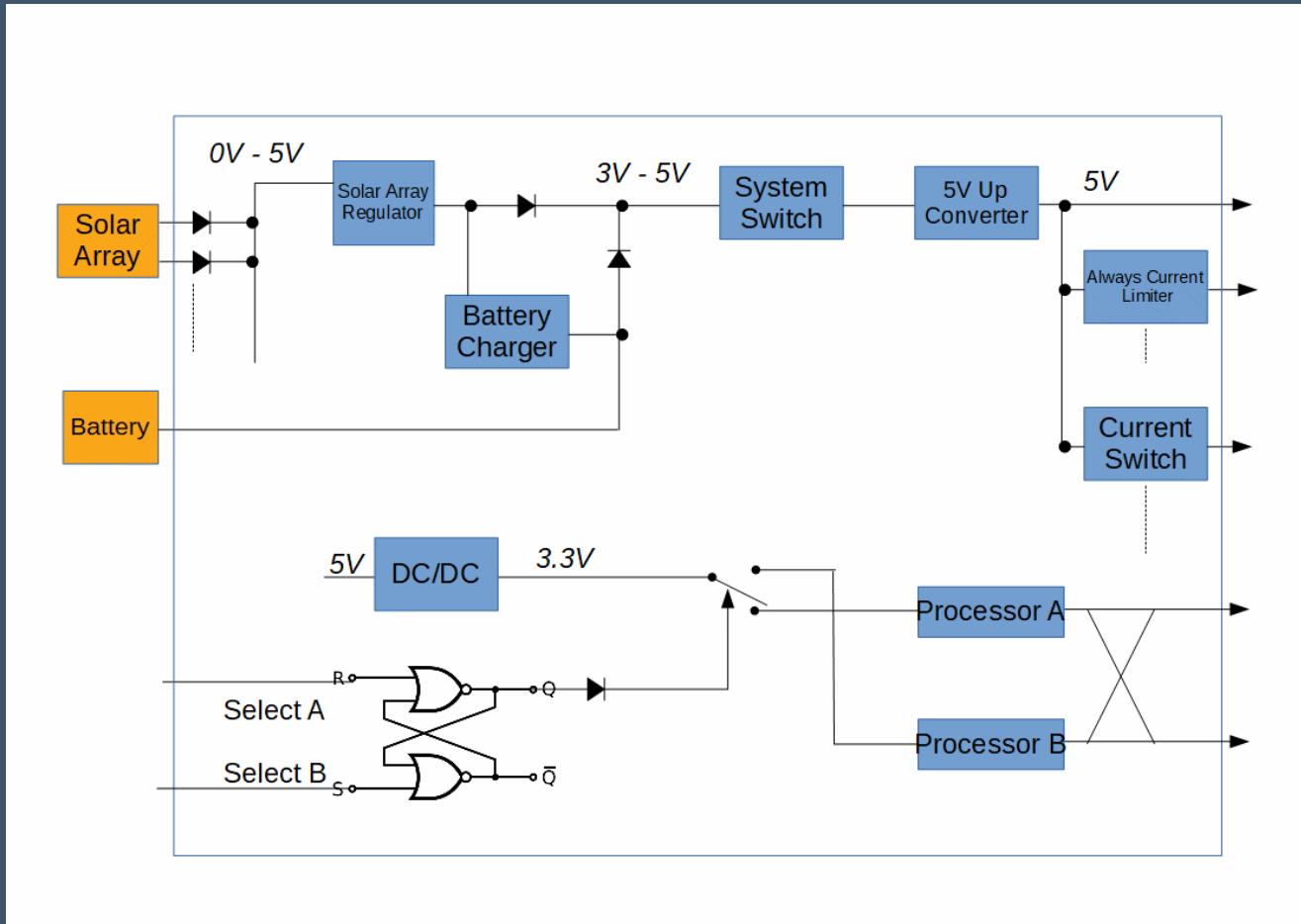
1U Structure



1U PCDU

Basic and Robust Power Supply System

Power Control and Distribution Unit



Get on Board

Communication Channels

A screenshot of the LibreCube website's home page. The header includes the LibreCube logo and the tagline "open source space and earth exploration". The navigation bar has links for Home, Blog, Repos, Activities, Community, Wiki, and Contact. The main content area is titled "HOME" and contains the following text:

LibreCube – Open Source Space and Earth Exploration

At LibreCube our vision is to **enable everyone** to get involved in **building systems for exploration** of near and remote locations using open source hardware and software. We believe that **discovering new worlds** and getting **scientific insights** should be a matter to **all humankind**.

LibreCube provides the means to it. Be part of it!

Below this text are three icons: a gear, an open book, and a server rack.

Open Source	Free and Open Standards	Reference Architecture
Everything we do at LibreCube is made available to the public as free and open source. And we only use free and open source tools – this way, really everyone can get involved!	We rely on proven and tested standards for our system designs, with preference to standards from the space domain.	Defining a generic architecture of system of systems that have standardized interfaces makes it possible to combine and reuse elements for various applications.

Chat



info@librecube.org