



cote

Software Simulation of the Space Environment

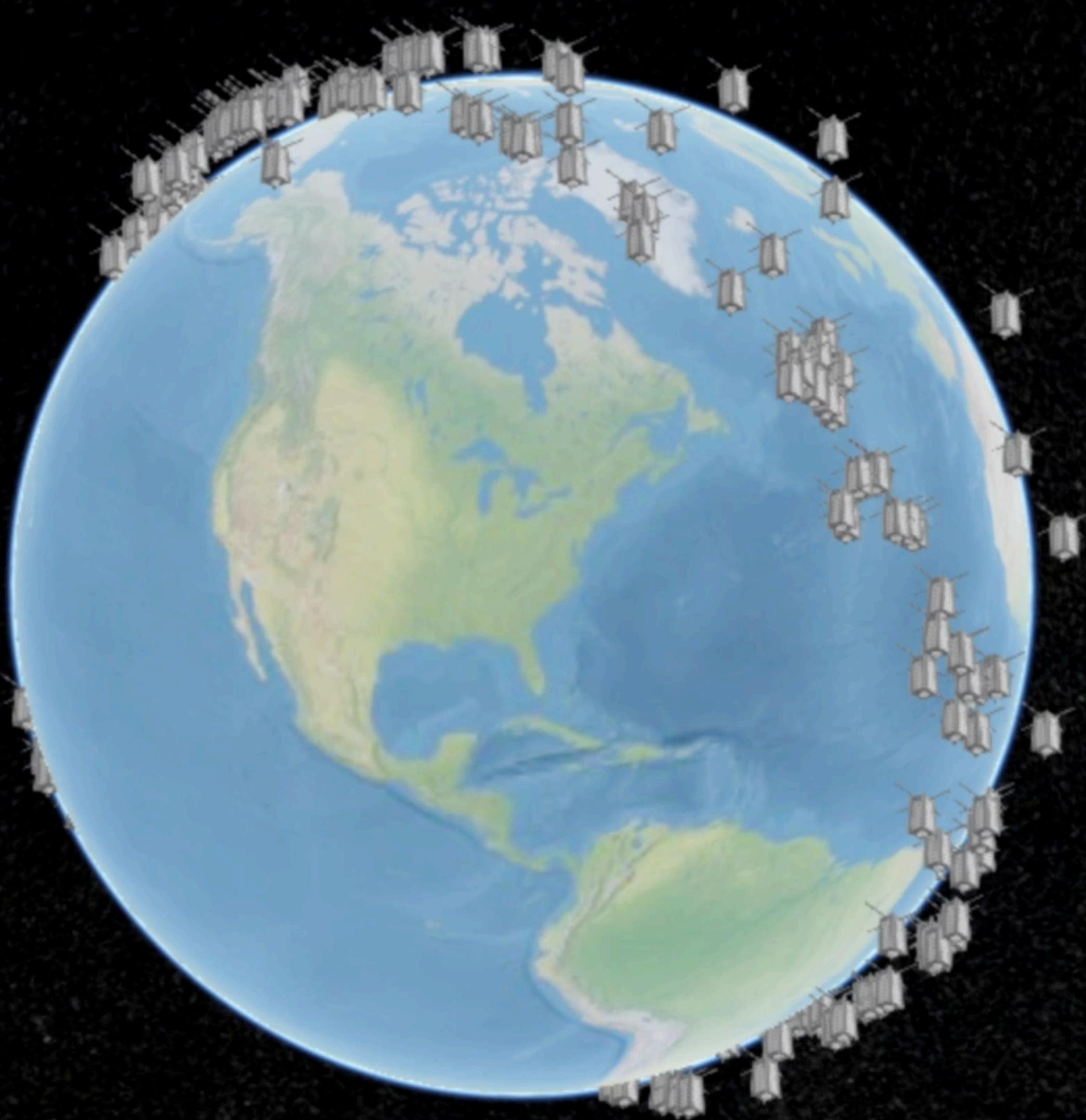
Brad Denby, bdenby@cmu.edu

Advisor: Brandon Lucia, blucia@andrew.cmu.edu

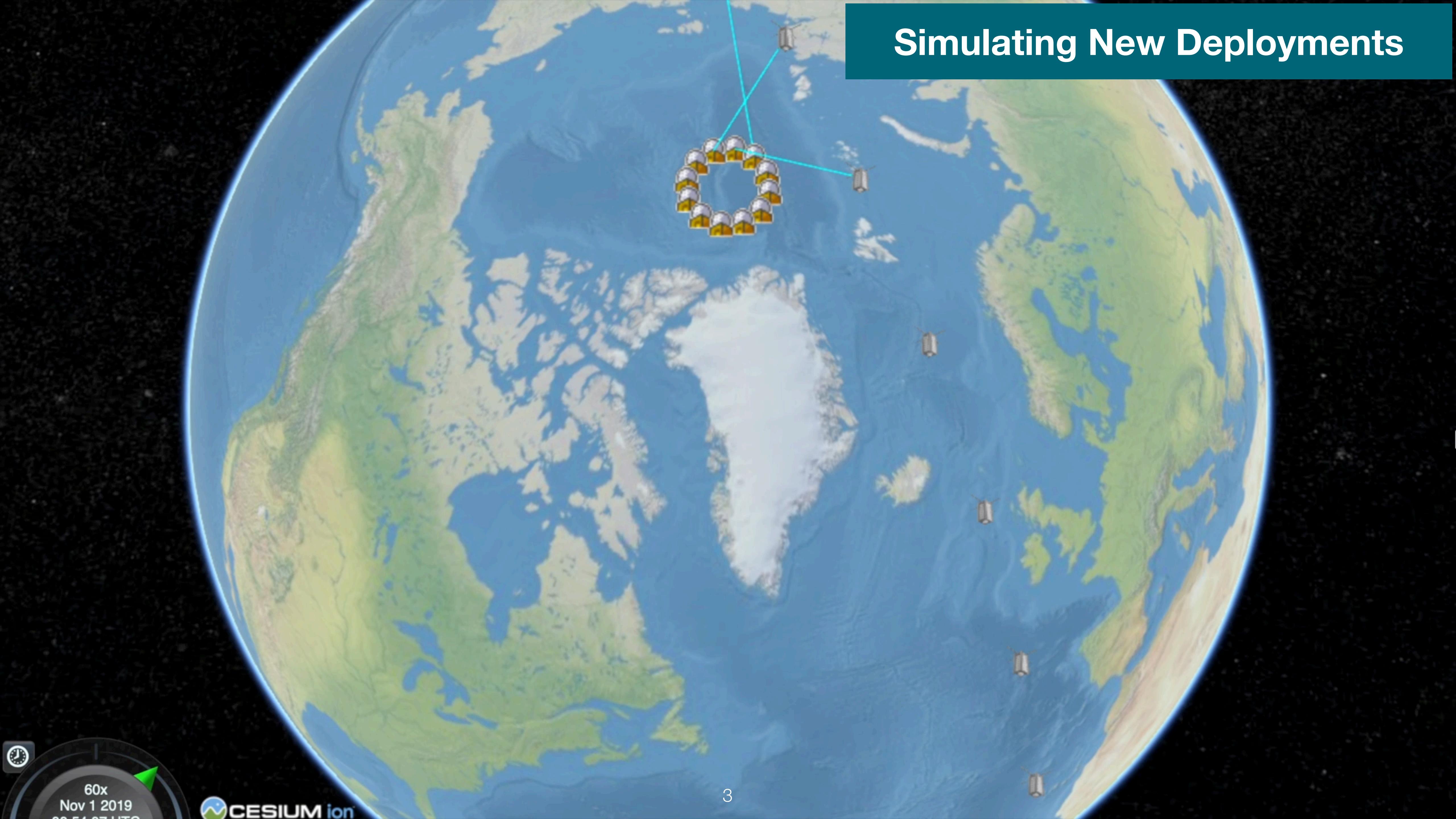
<https://github.com/CMUAbstract/cote>

Carnegie Mellon University

Simulating Existing Constellations



Simulating New Deployments



Overview

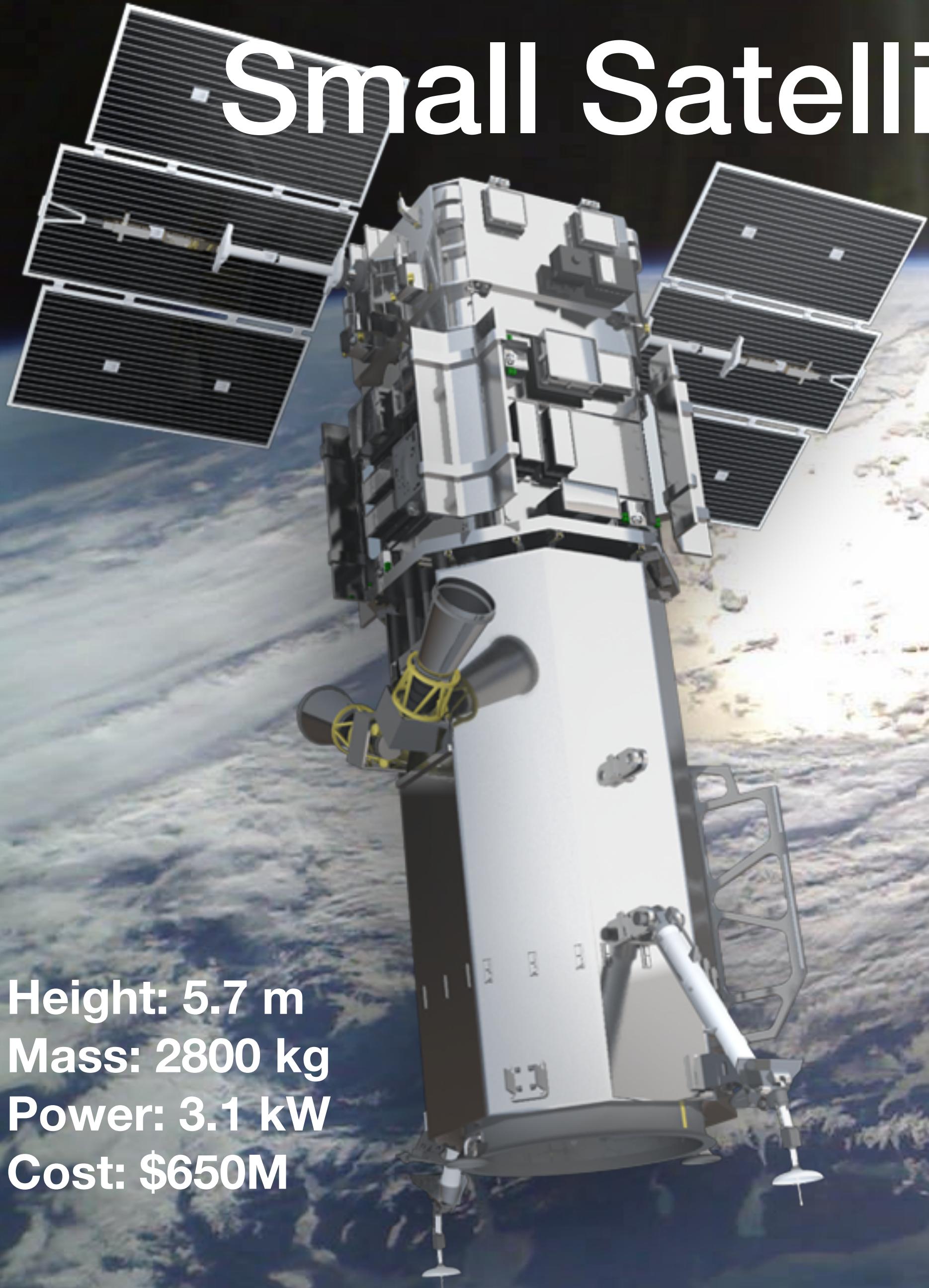
Background and Related Work

Components of Space Simulation

The cote Workflow

Example Programs in cote

Small Satellites Support Innovation



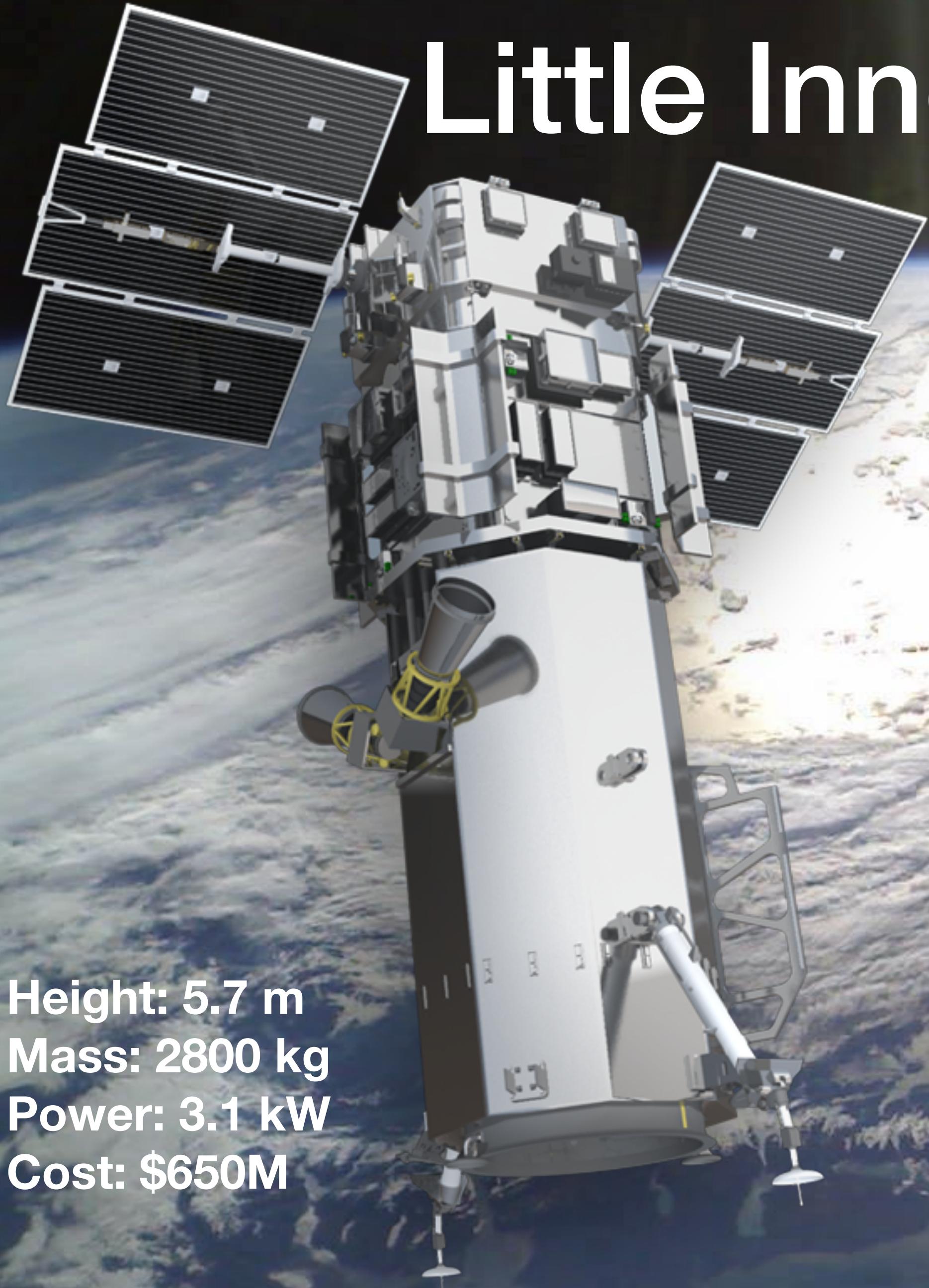
Height: 5.7 m
Mass: 2800 kg
Power: 3.1 kW
Cost: \$650M

Height: 0.3 m
Mass: 4 kg
Power: 7.1 W
Cost: \$65k



•
Pocketqubes and
Chip-scale Satellites

Little Innovation in CONOPS



Height: 5.7 m
Mass: 2800 kg
Power: 3.1 kW
Cost: \$650M

Command and Control

All Devices Share
a Bent-pipe CONOPS

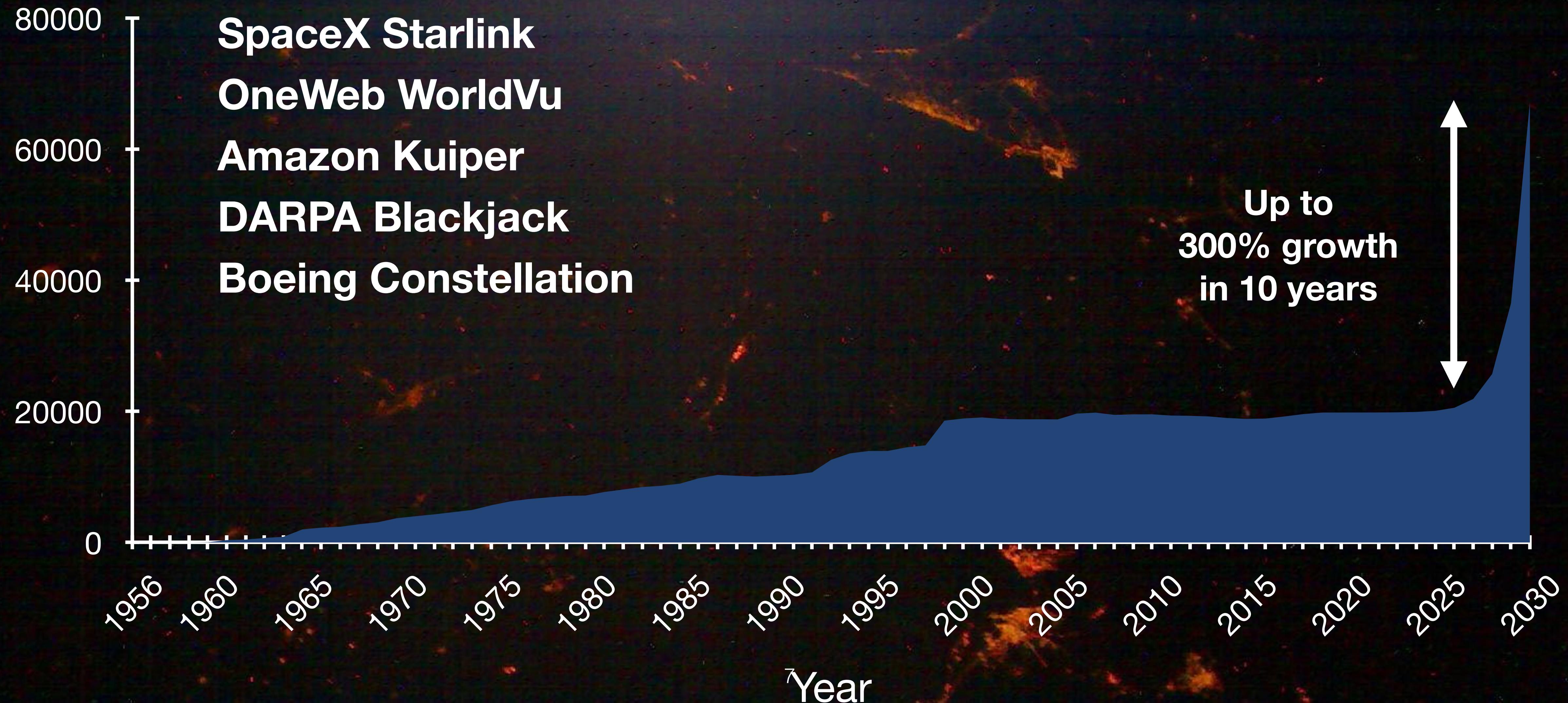
Height: 0.3 m
Mass: 4 kg
Power: 7.1 W
Cost: \$65k



Pocketqubes and
Chip-scale Satellites

CONOPS Must Change

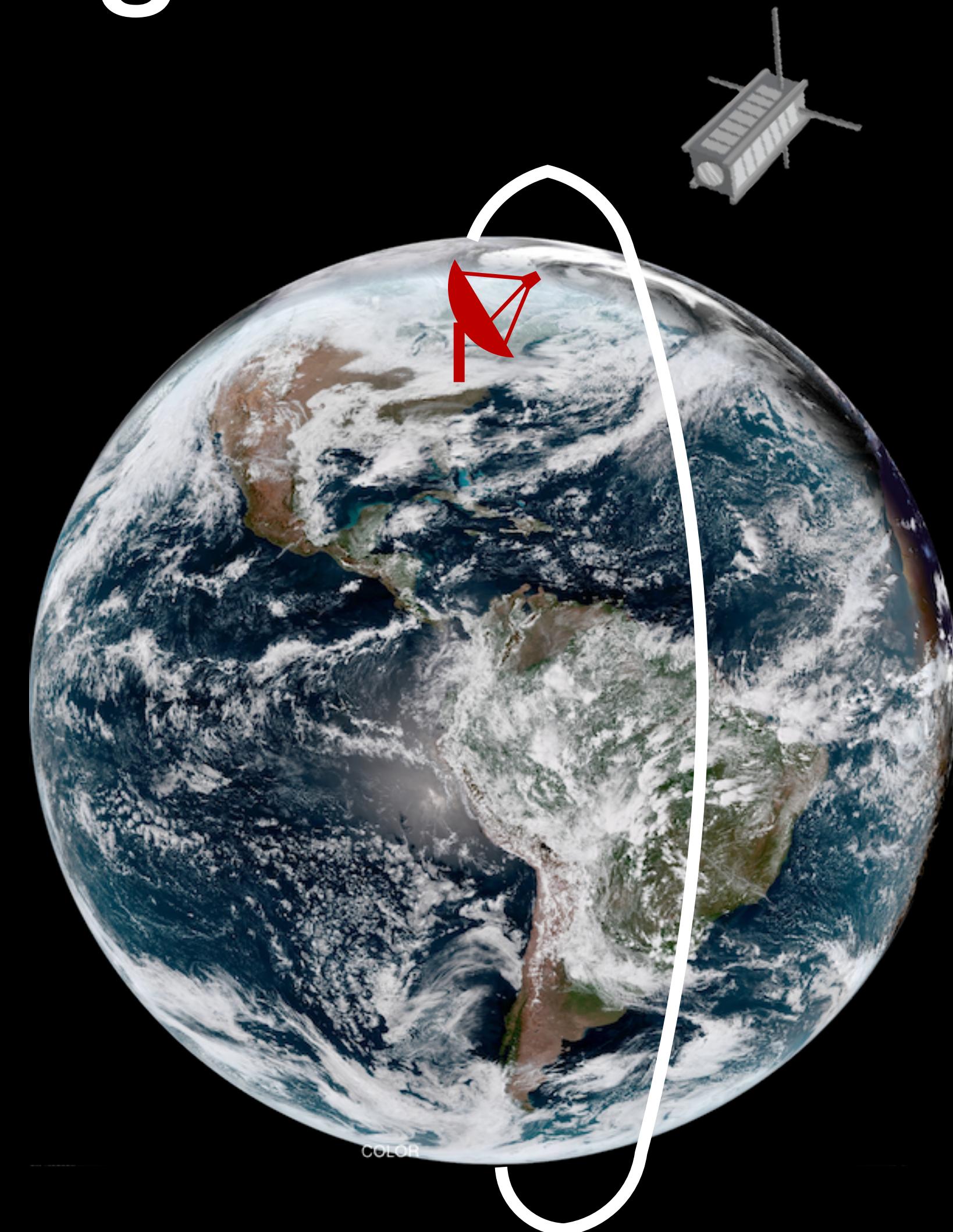
Satellites Orbiting Earth



Downlinking Data Does Not Scale

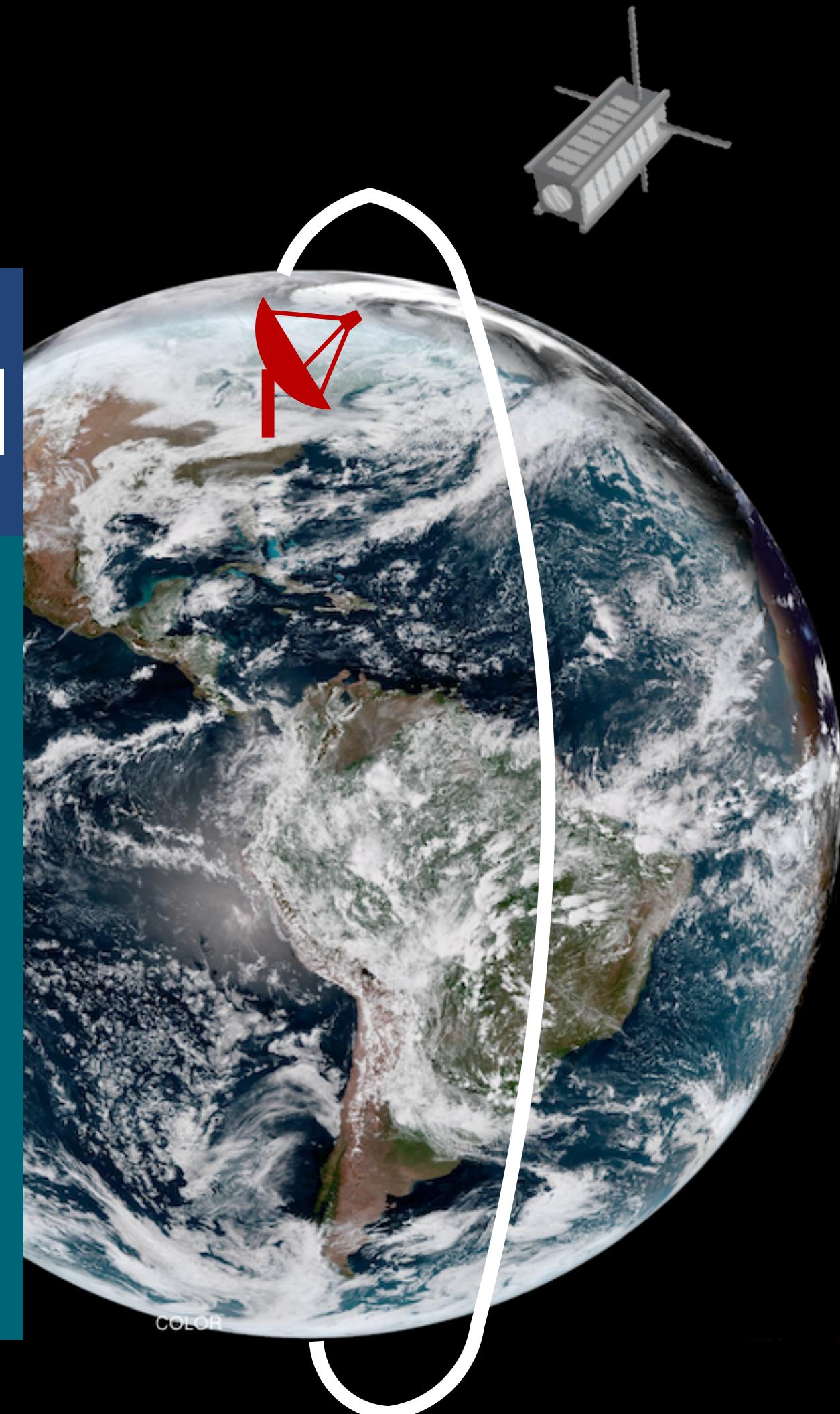


Downlinking Data Does Not Scale



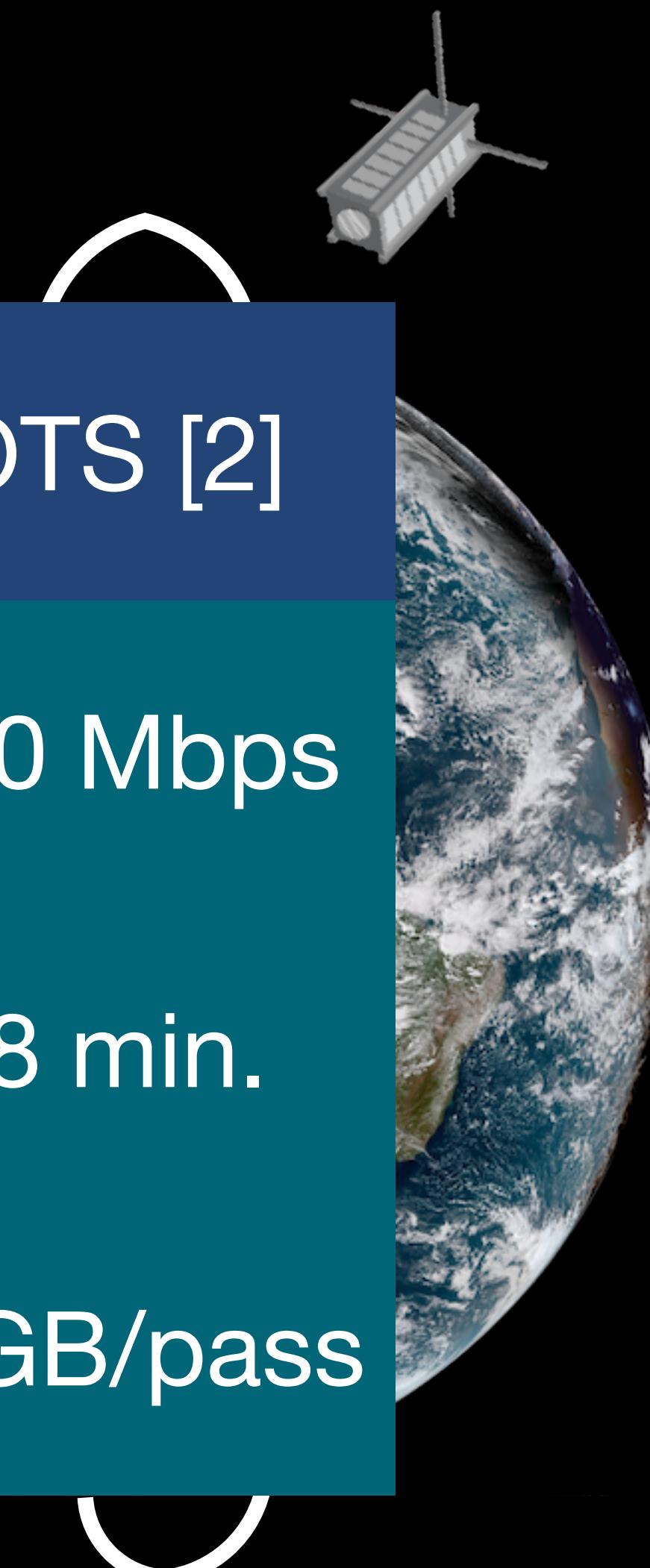
Downlinking Data Does Not Scale

Enterprise [1]	
Bitrate	200 Mbps
TX Duration	<10 min.
Data Down	15 GB/pass



[1] Dove High Speed Downlink System

Downlinking Data Does Not Scale

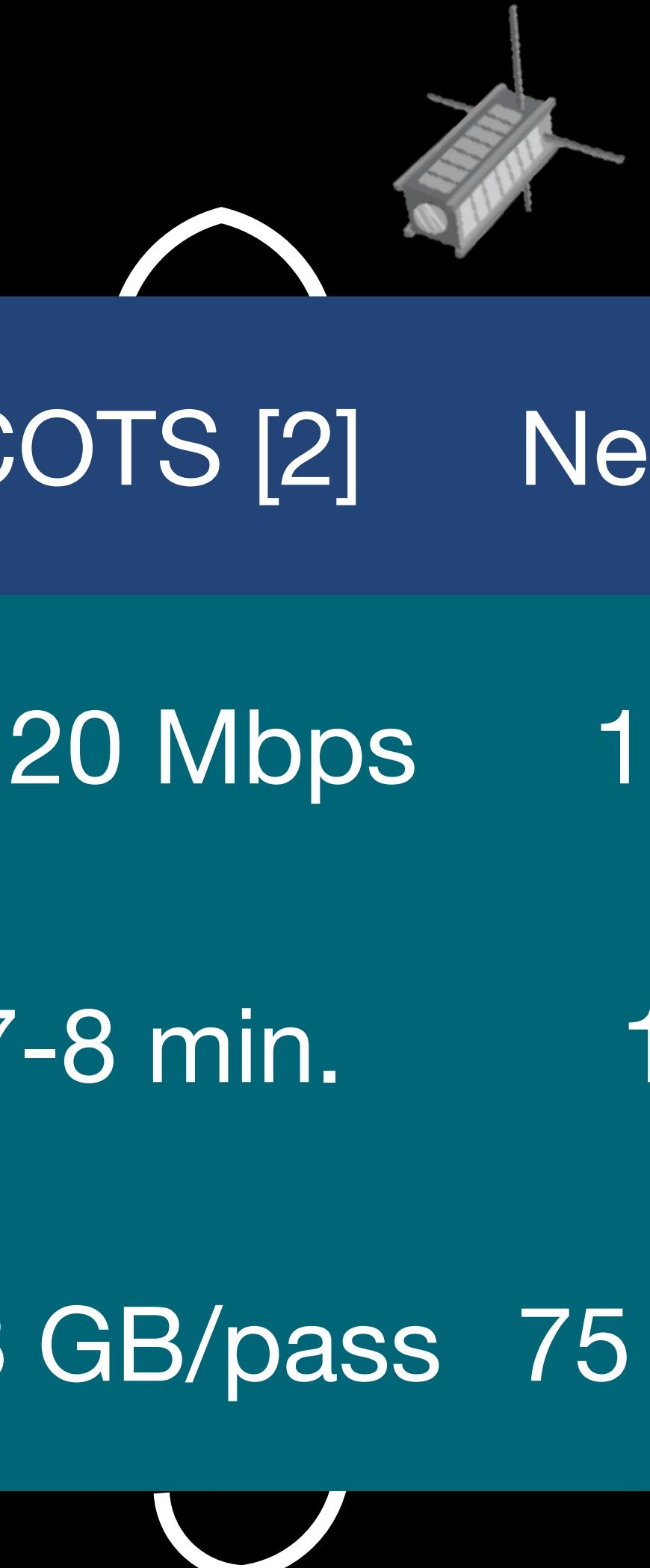


Enterprise [1] COTS [2]	
Bitrate	200 Mbps <120 Mbps
TX Duration	<10 min. 7-8 min.
Data Down	15 GB/pass 5.3 GB/pass

[1] Dove High Speed Downlink System

[2] cote Simulator: <https://github.com/cmuabstract/cote>

Downlinking Data Does Not Scale



	Enterprise [1]	COTS [2]	Near Future
Bitrate	200 Mbps	<120 Mbps	1 Gbps?
TX Duration	<10 min.	7-8 min.	10 min.
Data Down	15 GB/pass	5.3 GB/pass	75 GB/pass

[1] Dove High Speed Downlink System

[2] cote Simulator: <https://github.com/cmuabstract/cote>

Downlinking Data Does Not Scale

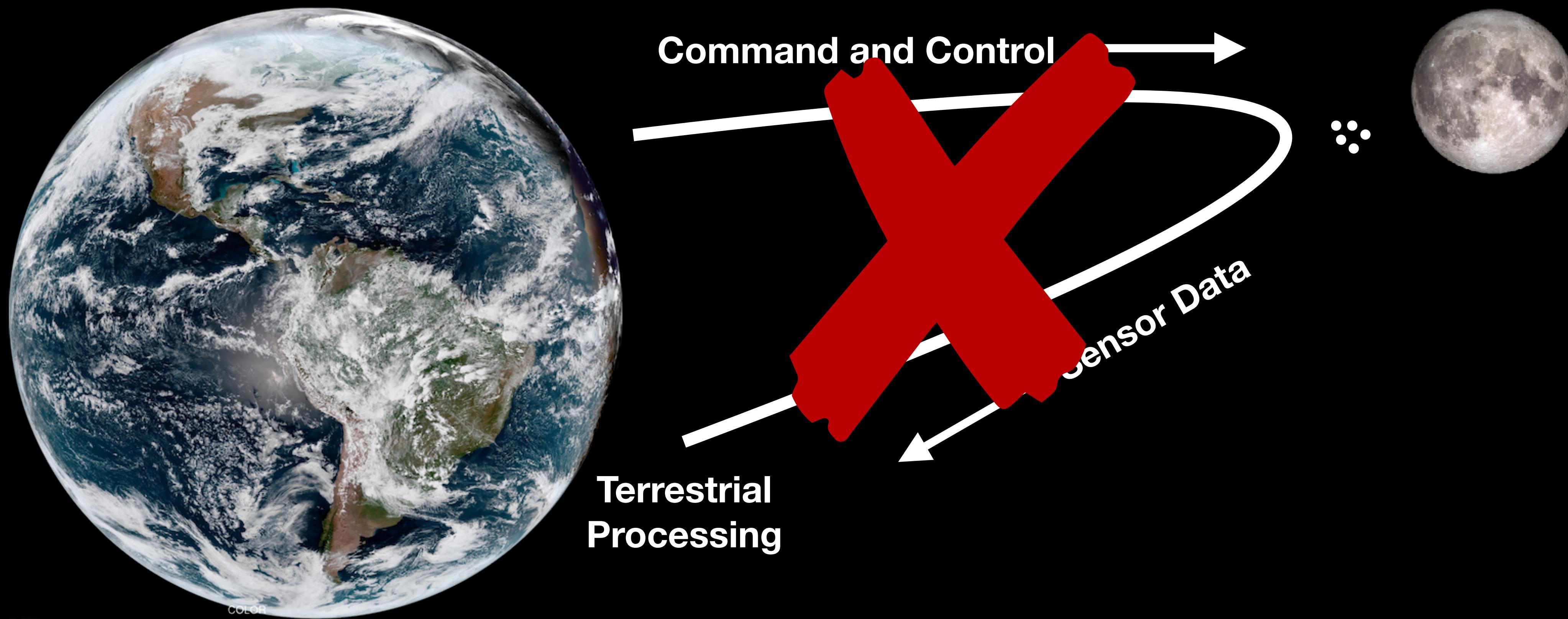


	Enterprise [1]	COTS [2]	Near Future	Mega Constellation
Bitrate	200 Mbps	<120 Mbps	1 Gbps?	No Change
TX Duration	<10 min.	7-8 min.	10 min.	↓(contention)
Data Down	15 GB/pass	5.3 GB/pass	75 GB/pass	Less per sat.

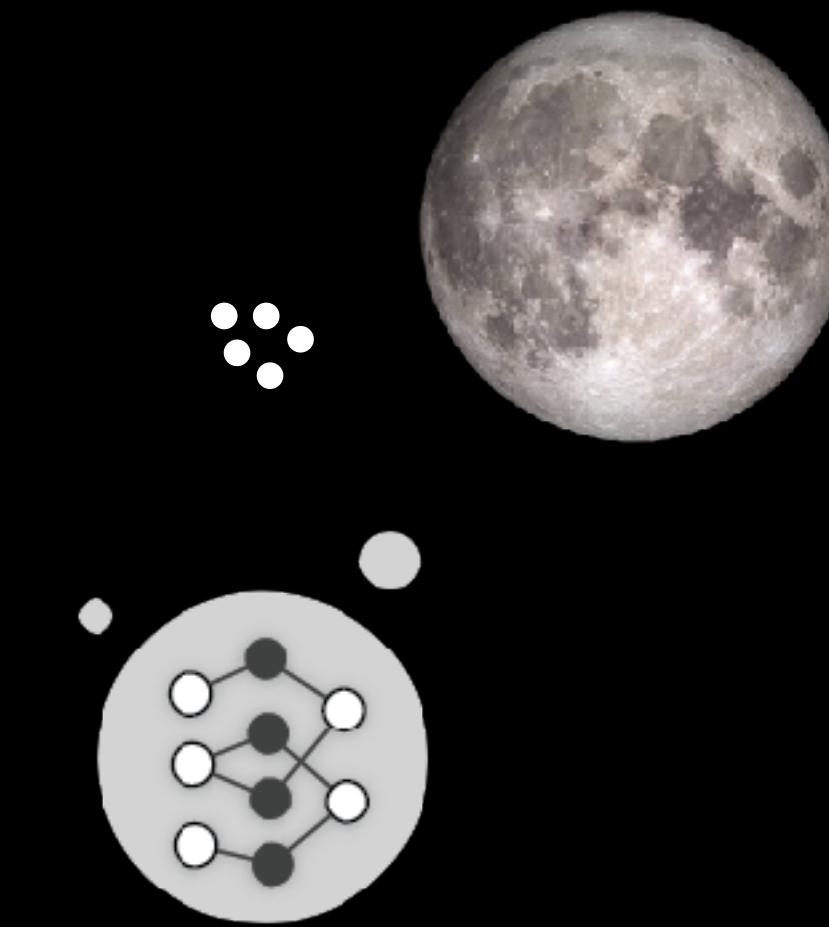
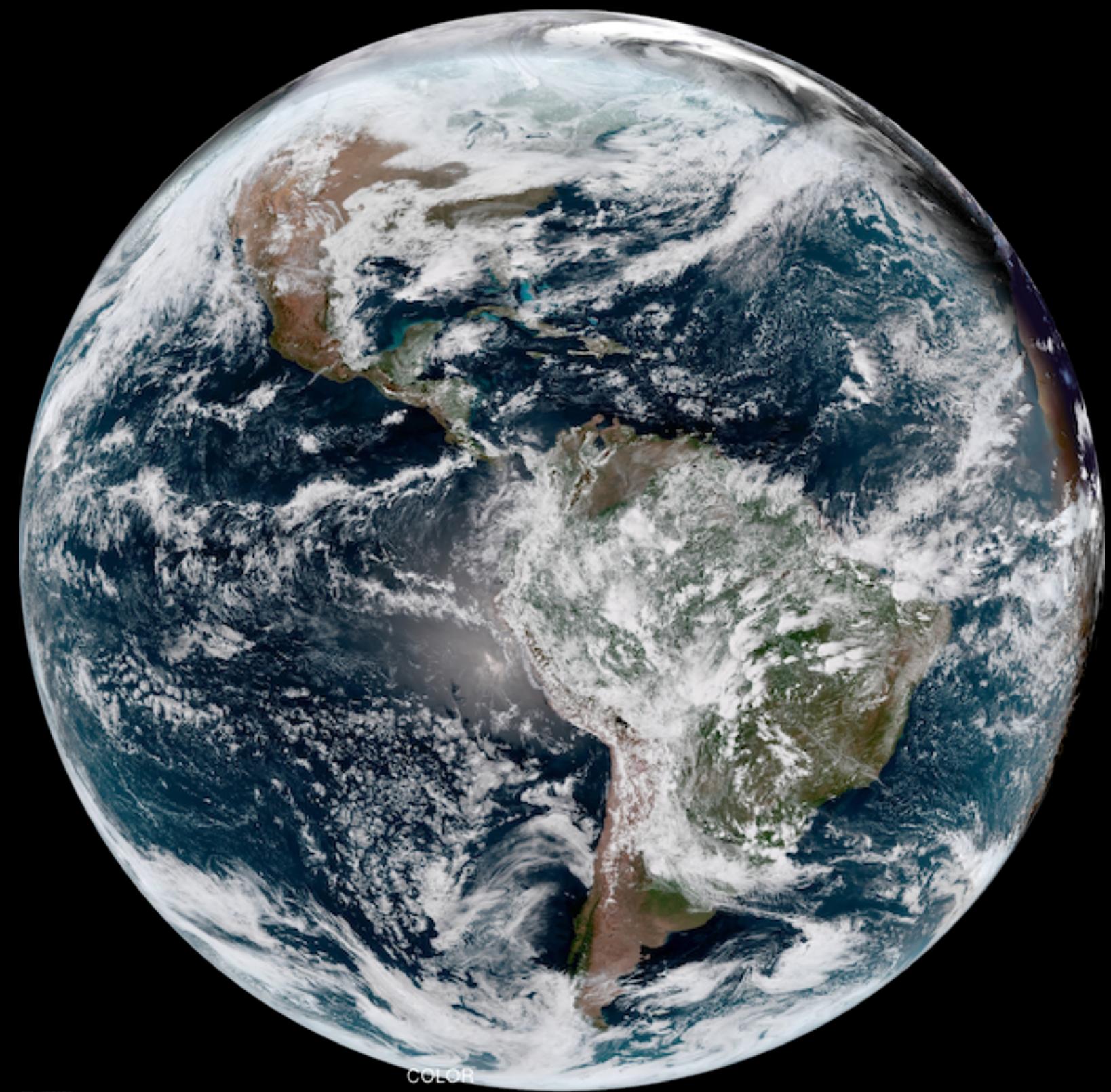
[1] Dove High Speed Downlink System

[2] cote Simulator: <https://github.com/cmuabstract/cote>

Reducing Reliance on Bent Pipes

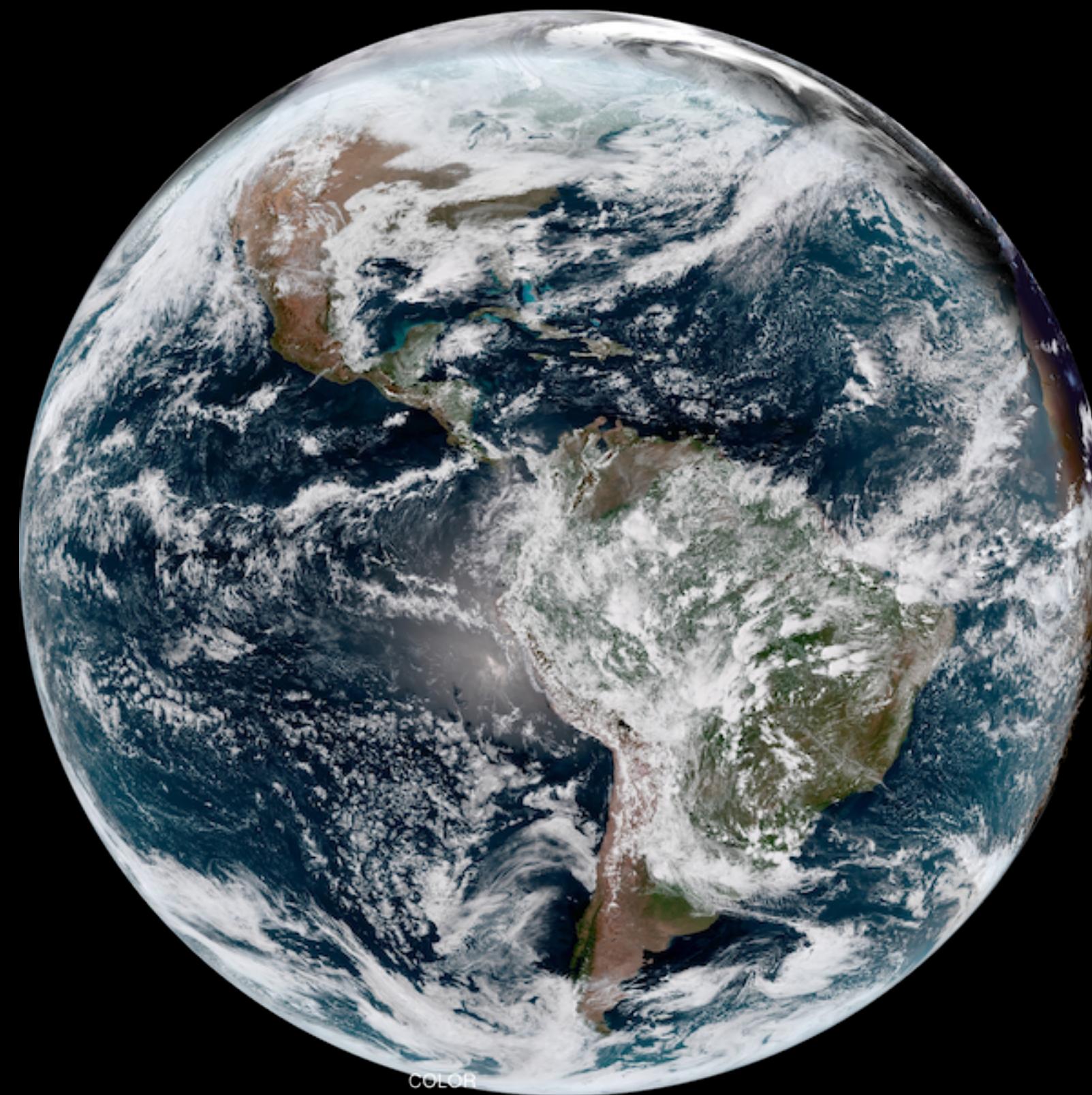


Edge Computing: A Scalable Alternative

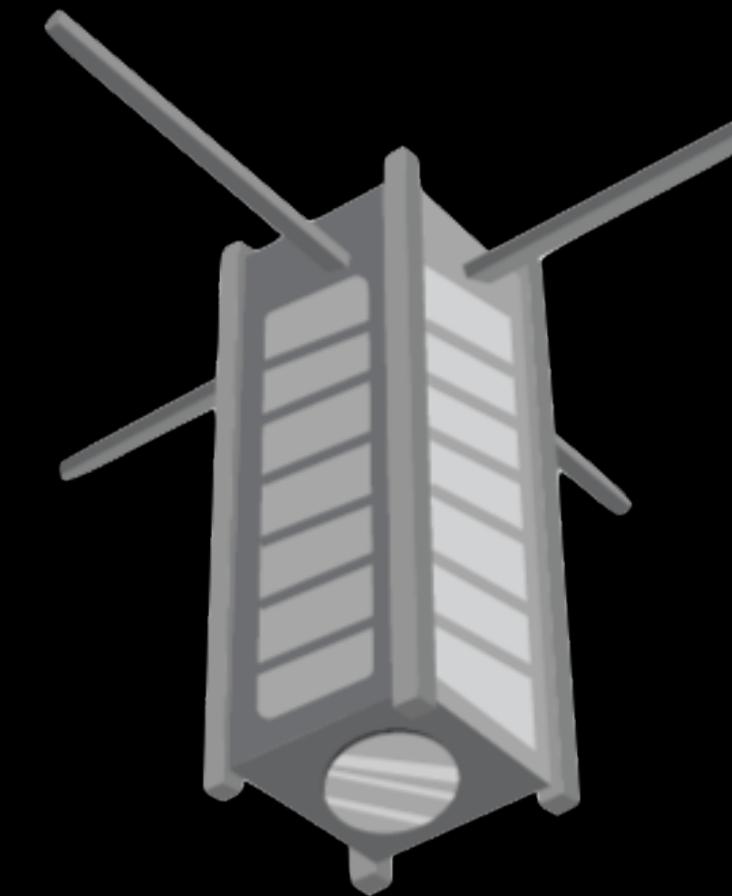


**Computing
at the Edge**

Edge Computing: A Scalable Alternative



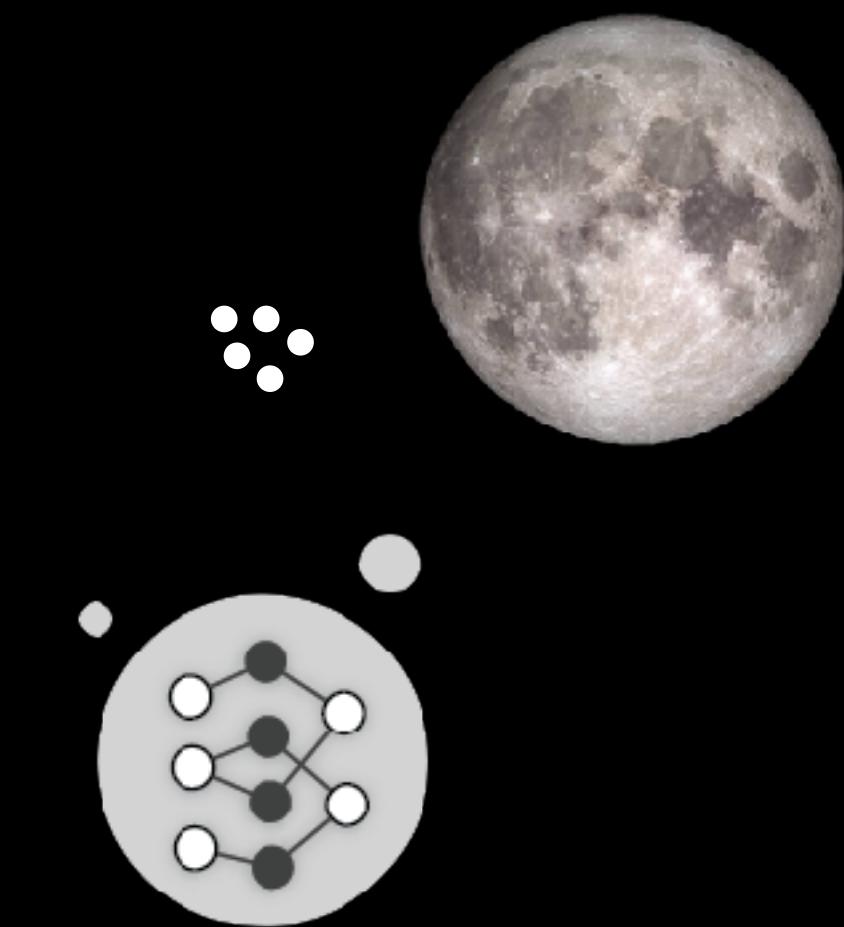
Existing Systems



Communication

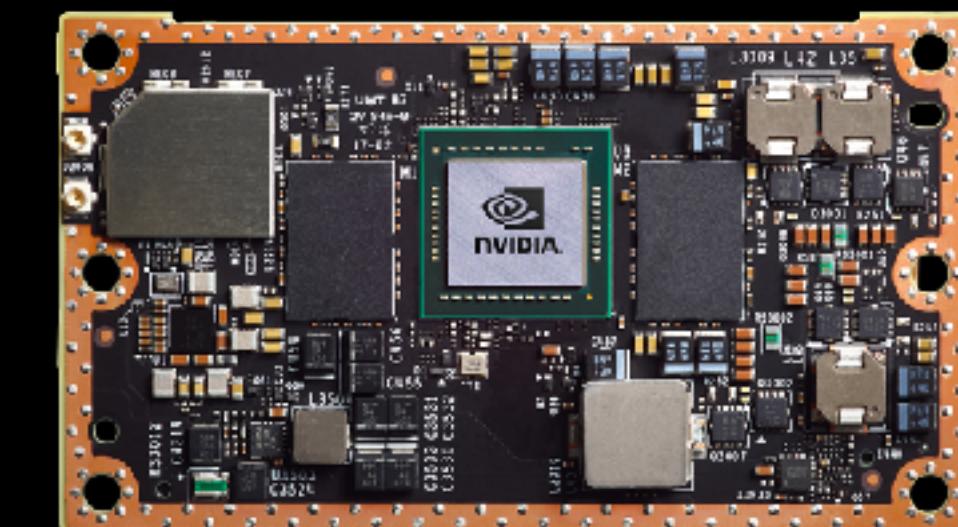
Guidance,
Navigation,
& Control

Sensors



Edge Computing in Space

e.g. Embedded Compute Module



Computing
at the Edge

Overview

Background and Related Work

Components of Space Simulation

The cote Workflow

Example Programs in cote

Computational Nanosatellites

Machine Learning in Space



Machine Learning Bridges the Gap

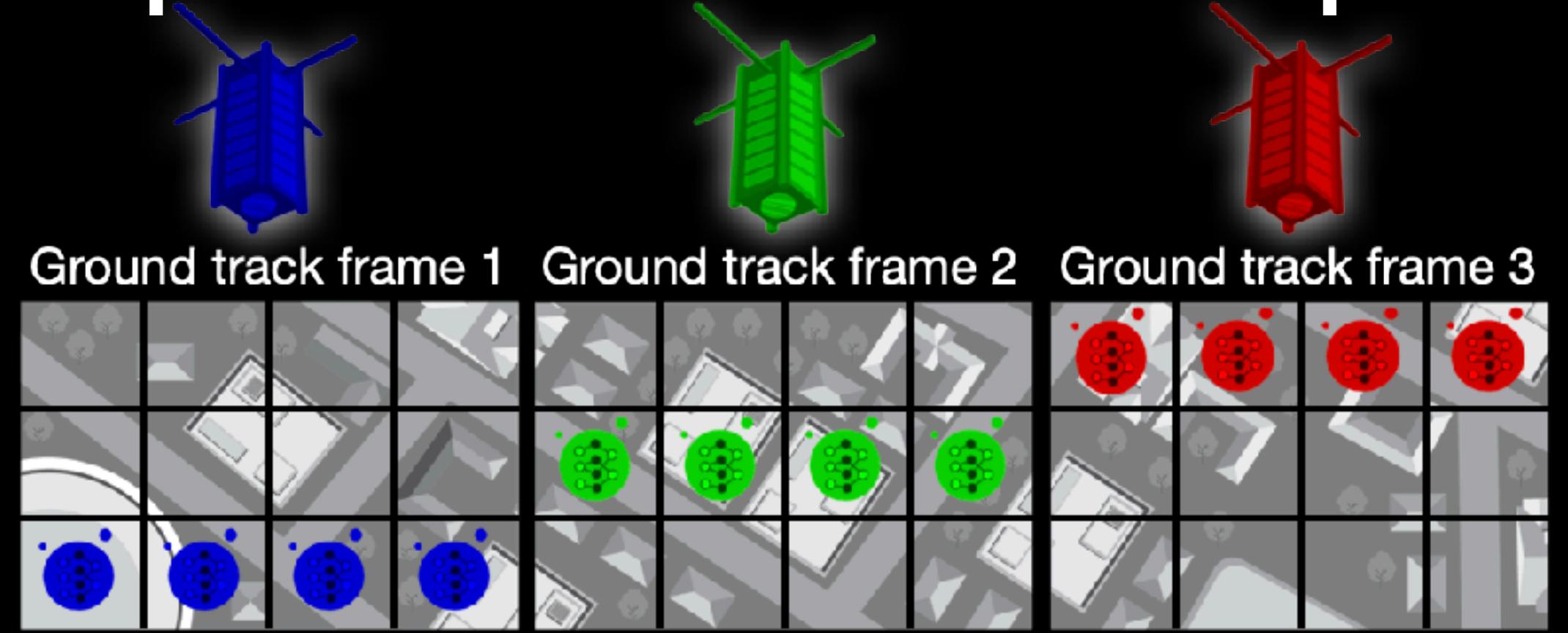
*Orbital Edge Computing: Machine Inference in Space
(CAL'19; B Denby, B Lucia)*

Computational Nanosatellites

Machine Learning in Space



Computational Nanosatellite Pipelines



Machine Learning Bridges the Gap

*Orbital Edge Computing: Machine Inference in Space
(CAL'19; B Denby, B Lucia)*

Distributing Computation across Devices

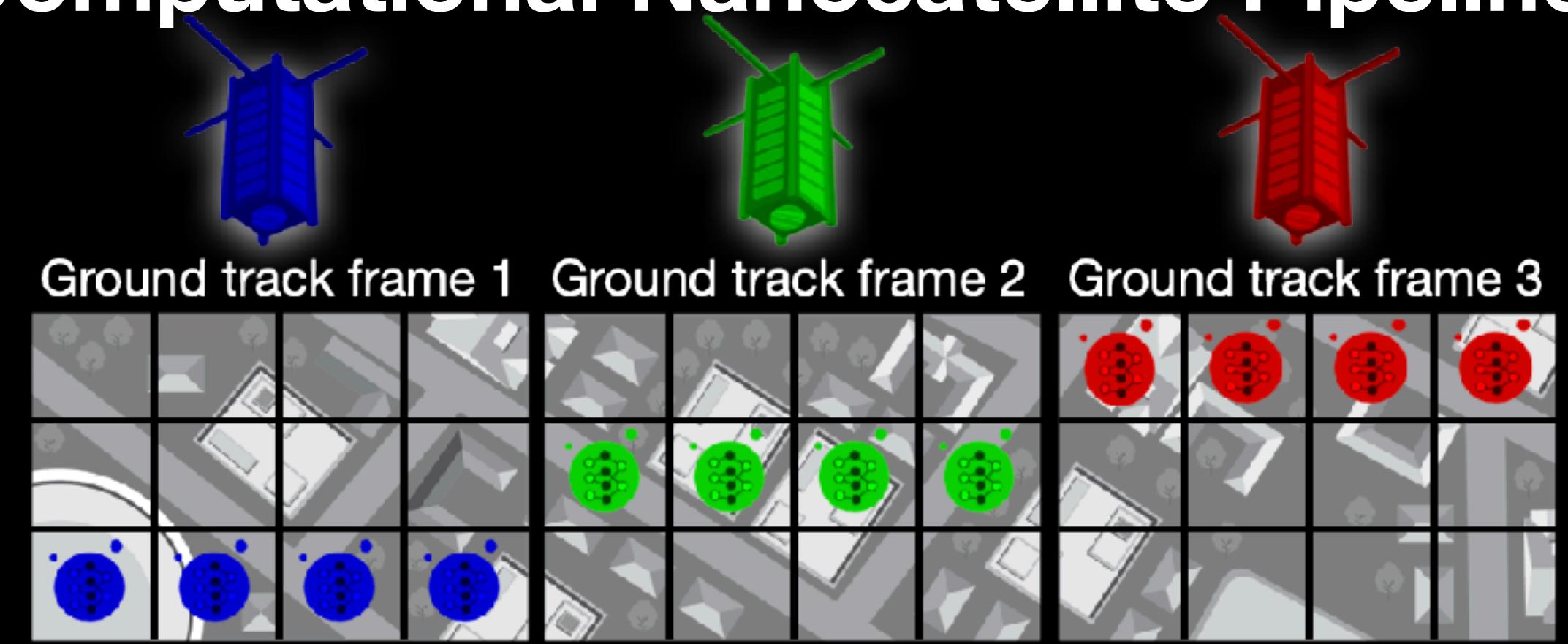
Orbital Edge Computing: Nanosatellite Constellations as a New Class of Computer System (ASPLOS'20; B Denby, B Lucia)

Computational Nanosatellites

Machine Learning in Space



Computational Nanosatellite Pipelines



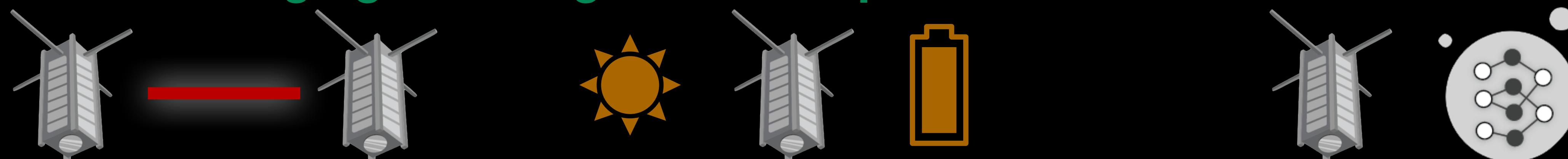
Machine Learning Bridges the Gap

*Orbital Edge Computing: Machine Inference in Space
(CAL'19; B Denby, B Lucia)*

Distributing Computation across Devices

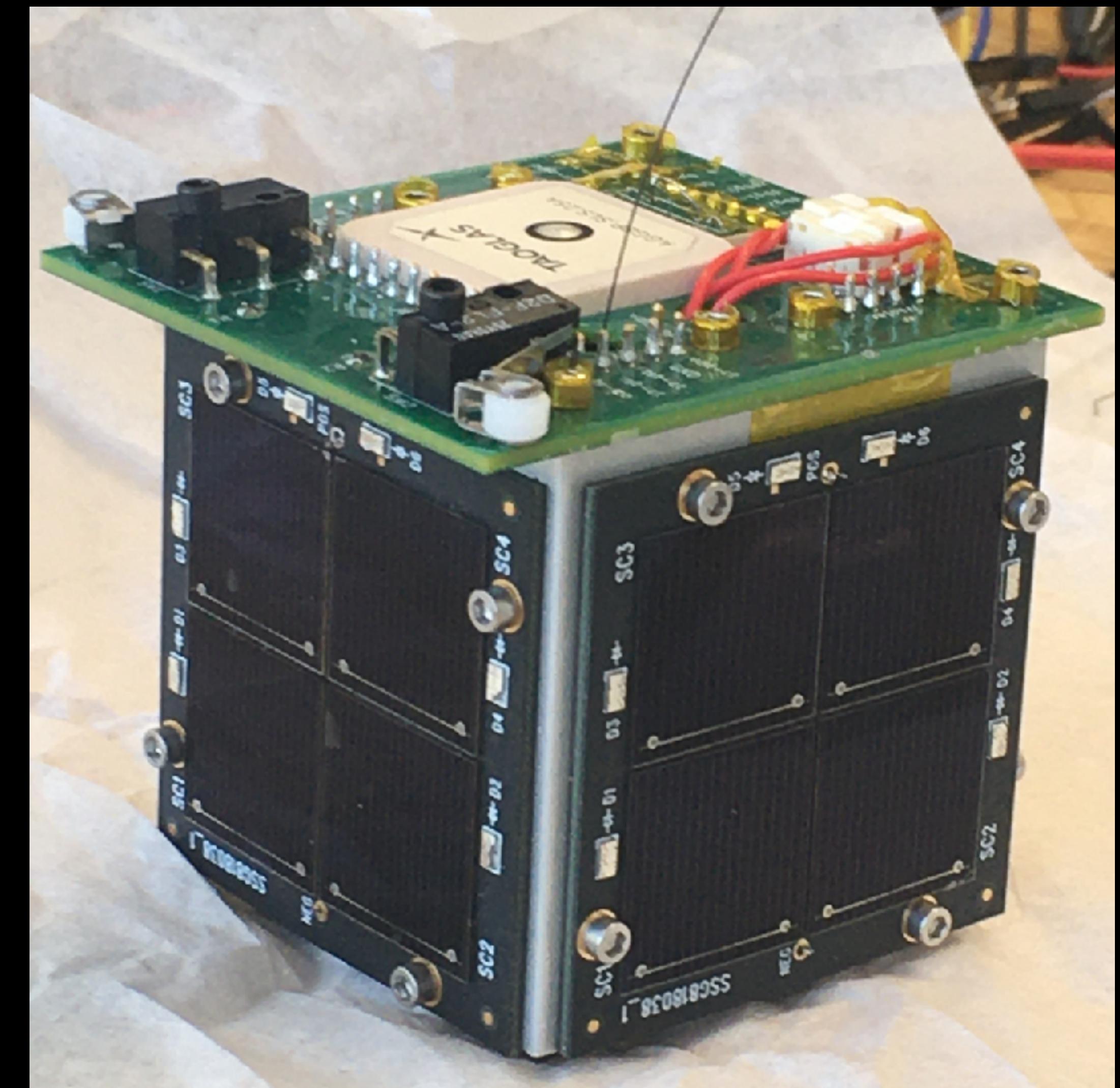
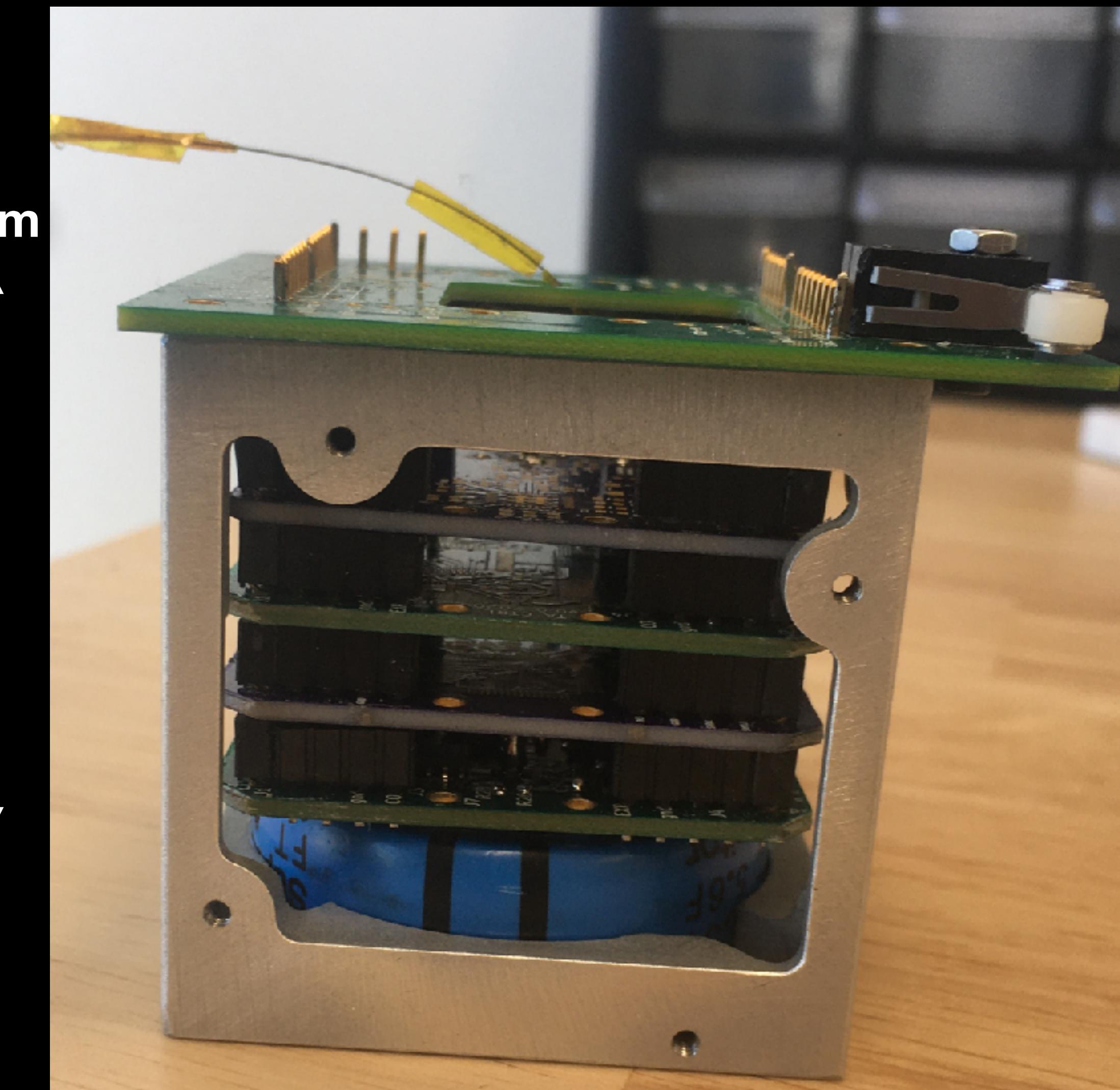
Orbital Edge Computing: Nanosatellite Constellations as a New Class of Computer System (ASPLOS'20; B Denby, B Lucia)

Emerging Challenges for Computational Nanosatellites

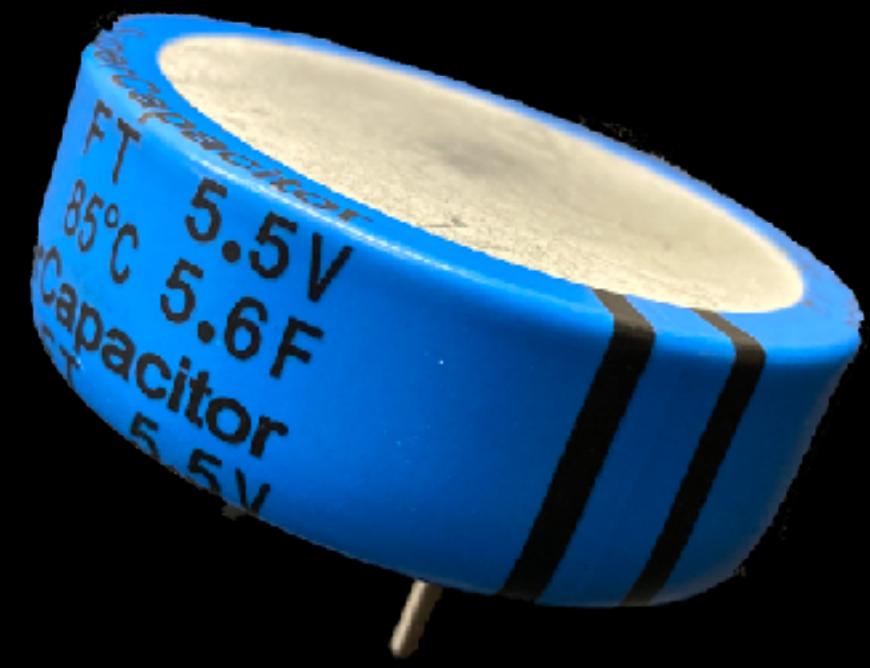
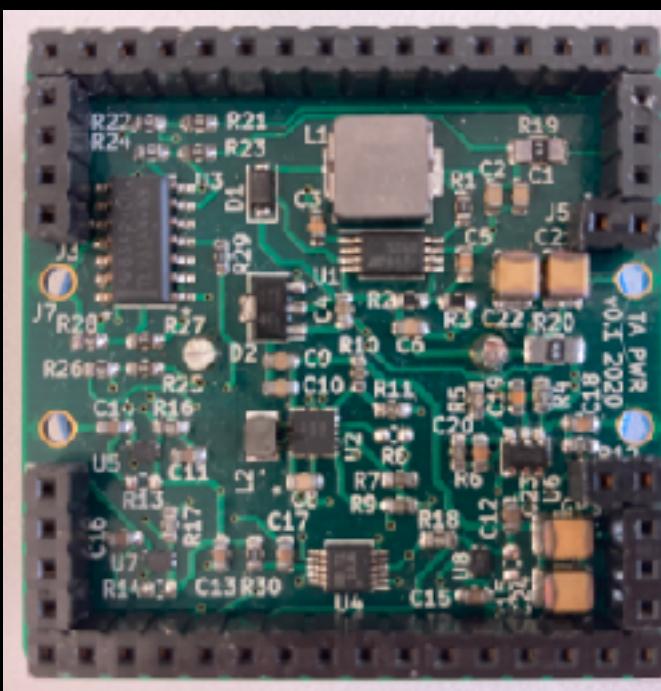
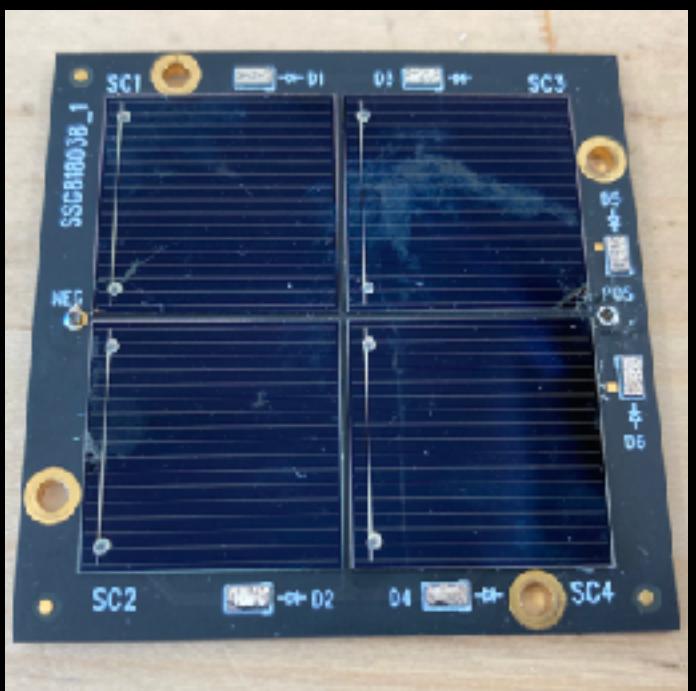


*Computational Nanosatellite Constellations: Opportunities and Challenges
(GetMobile'21; B Lucia, B Denby, Z Manchester, H Desai, E Ruppel, A Colin)*

Tartan Artibeus: Evaluation in Space

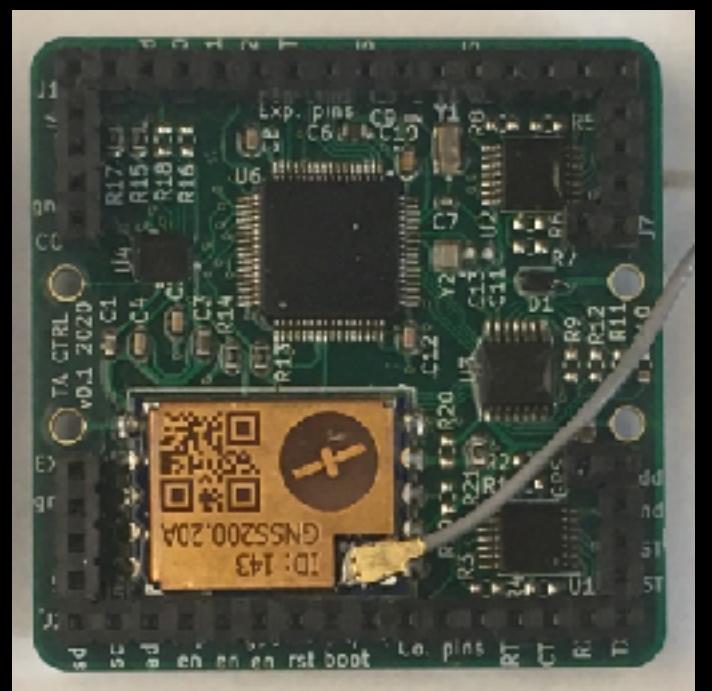
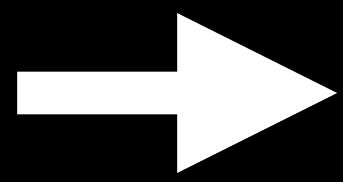
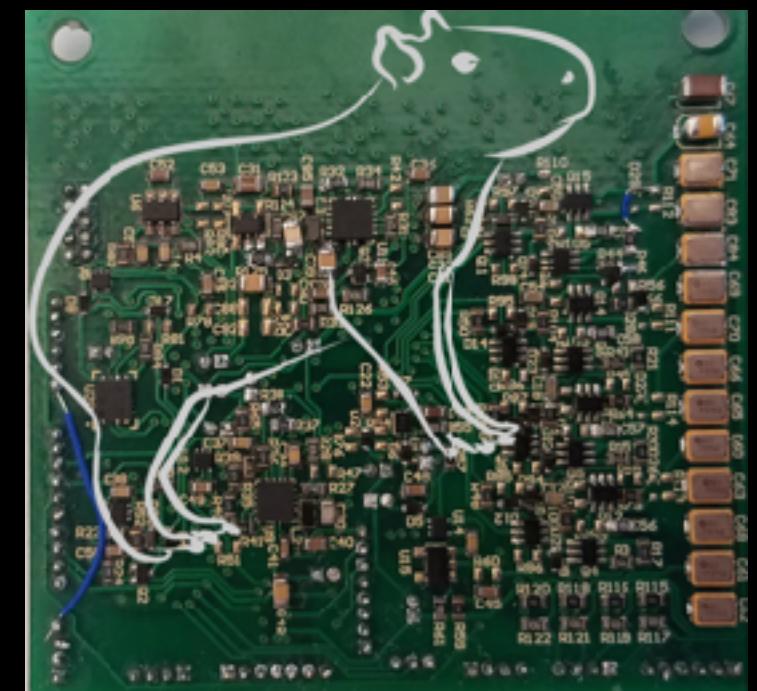


Tartan Artibeus: Evaluation in Space



Energy System

Command and Control



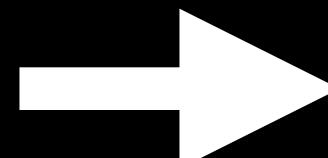
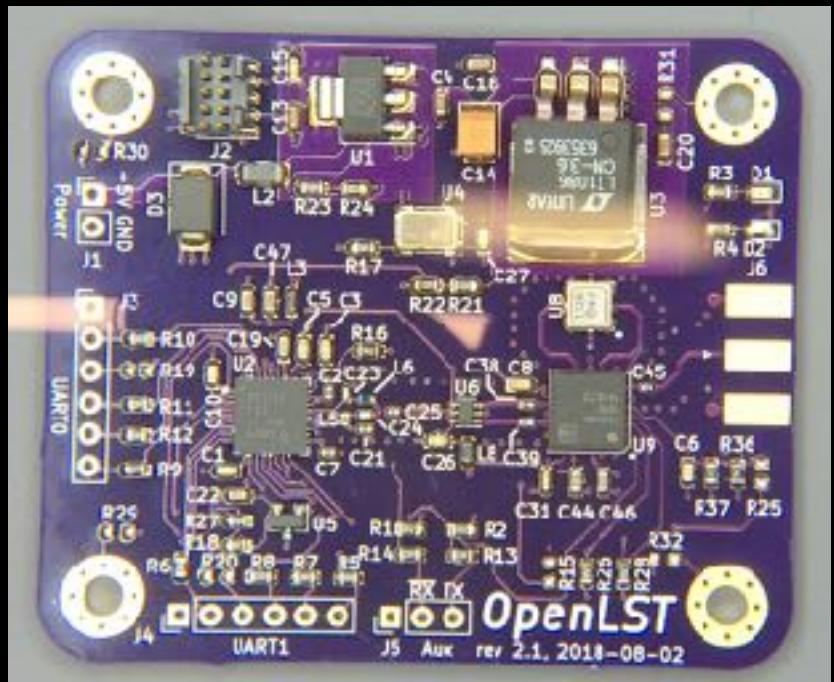
Capybara

Pocketqube C&DH

Tartan Artibeus: A Batteryless, Computational Satellite Research Platform

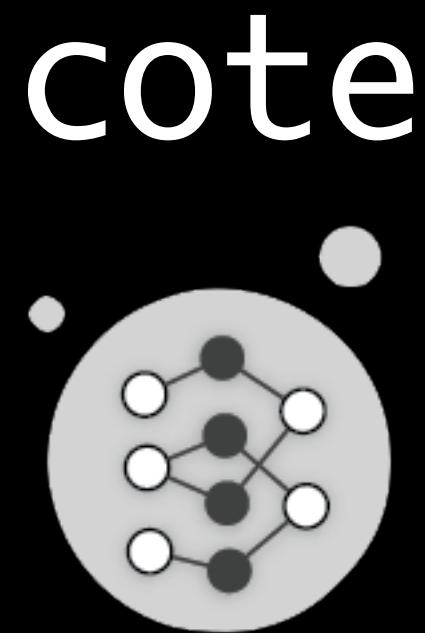
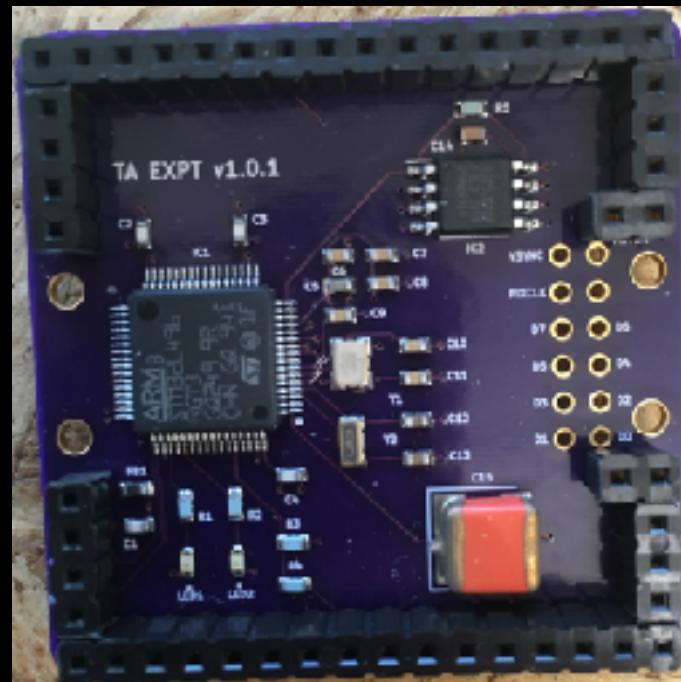
(*SmallSat'22; B Denby, E Ruppel, V Singh, S Che, C Taylor, F Zaidi, S Kumar, Z Manchester, B Lucia*)

Radio Communication



OpenLST

Computational Payload



Pocketqube Radio

Overview

Background and Related Work

Components of Space Simulation

The cote Workflow

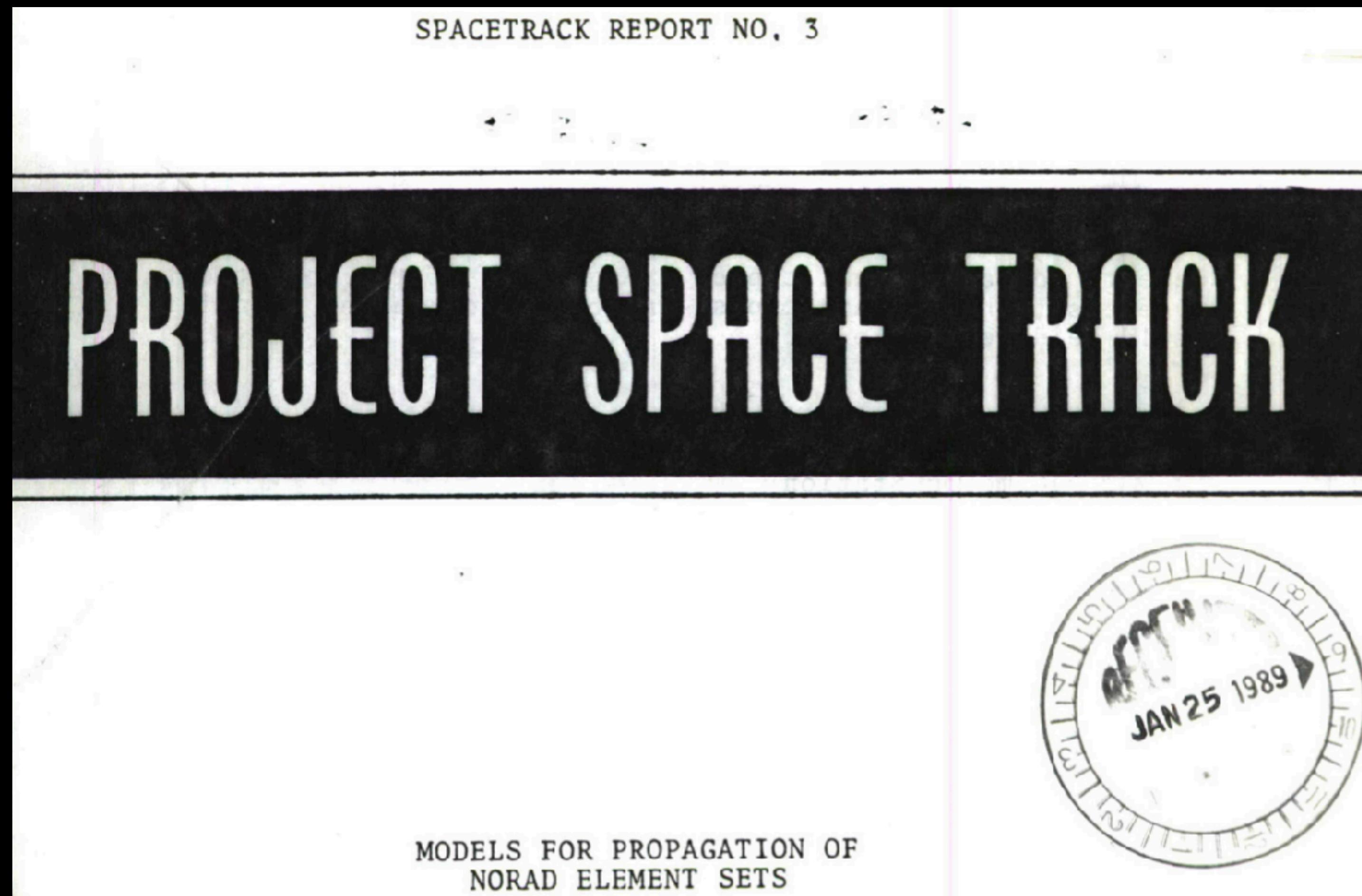
Example Programs in cote

Components of Space Simulation

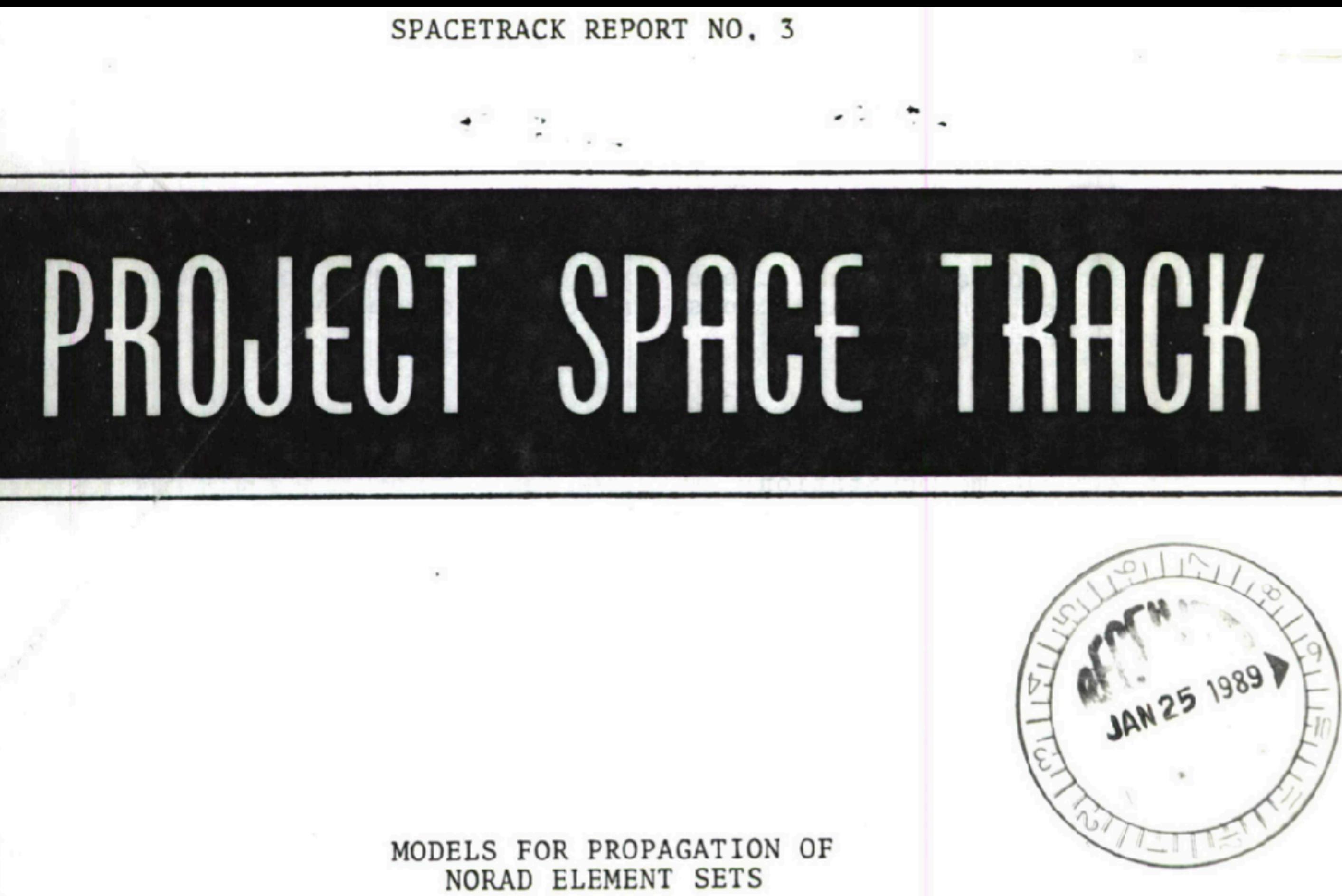
Orbital Mechanics



Orbital Mechanics



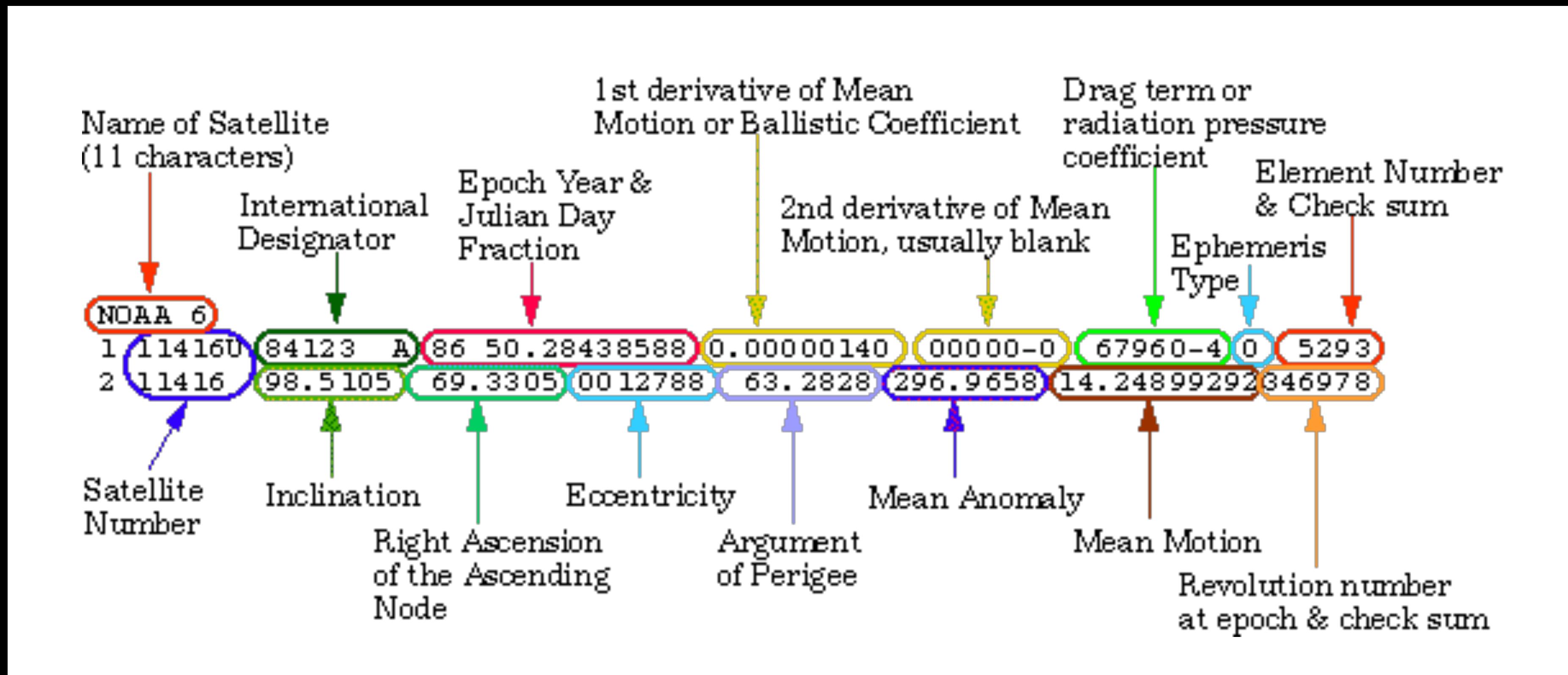
Orbital Mechanics



- SGP4: Simplified General Perturbations 4
 - Input: Date, Time, and Satellite TLE (two-line element set)
 - Output: Three-dimensional satellite position
- Benefits: Calculate the position of any publically-deployed satellite
- Drawbacks: Limited utility for modeling the position of prospective satellites

SGP4 Input: TLE (Two-line element set)

TLEs for all satellites are posted ~daily for download

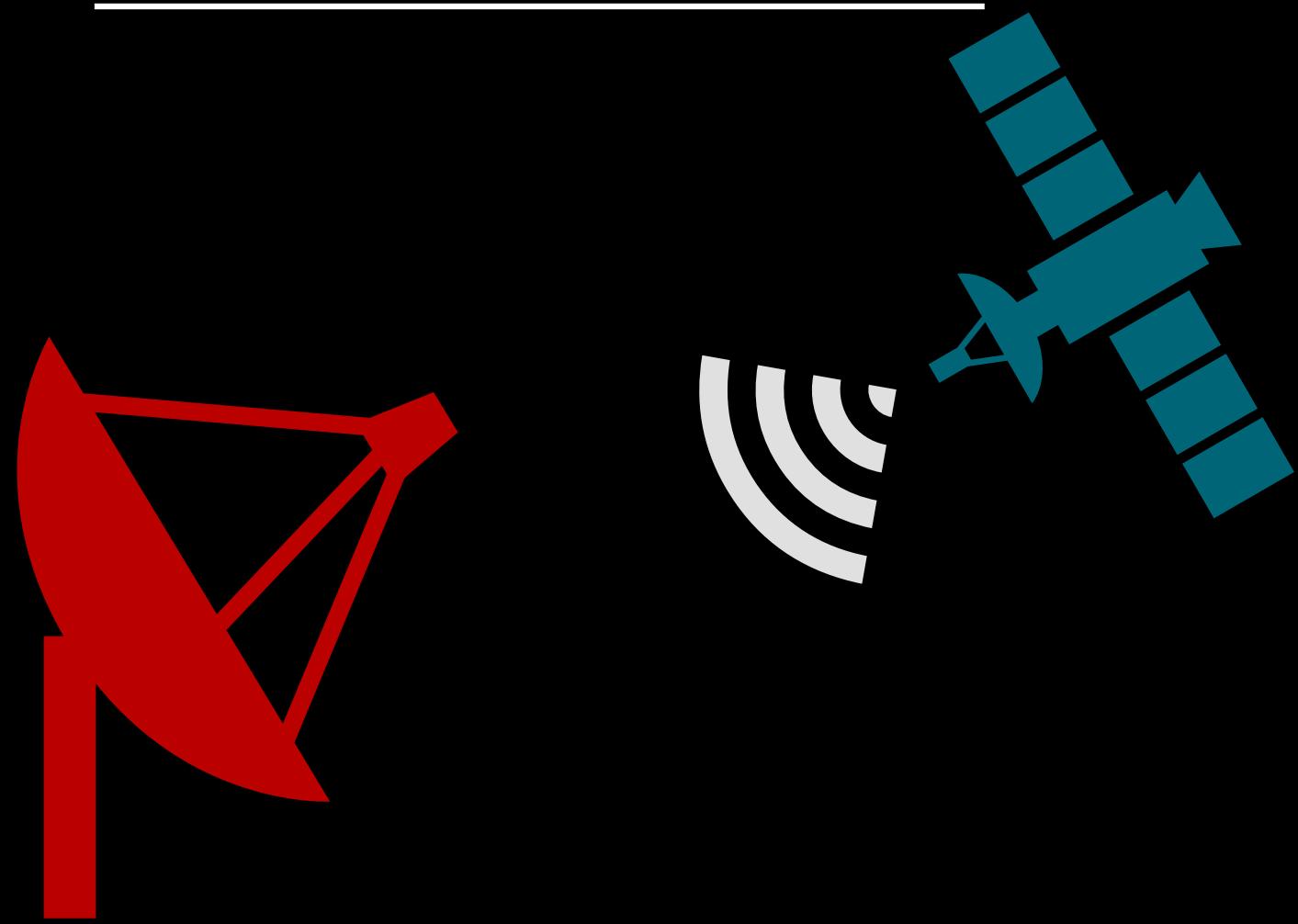


Components of Space Simulation

Orbital Mechanics



Communication



Satellite Communications

Space Mission Analysis and Design Third Edition

Edited by

Wiley J. Larson
United States Air Force Academy

and

James R. Wertz
Microcosm, Inc.

Coordination by

Douglas Kirkpatrick, United States Air Force Academy
Donna Klungle, Microcosm, Inc.

This book is published as part of the Space Technology Series, a cooperative activity of the United States Department of Defense and National Aeronautics and Space Administration.



Space Technology Library

Published Jointly by

Chapter 13

Communications Architecture

Fred J. Dietrich, Globalstar L.P.
Richard S. Davies, Stanford Telecommunications, Inc.

- 13.1 Communications Architecture
Communications Architecture Defined by Satellite-Ground Station Geometry; Communications Architecture Defined by Function; Criteria for Selecting Communications Architecture
- 13.2 Data Rates
TT&C; Data Collection; Data Relay
- 13.3 Link Design
Derivation of Link Equation; Link Design Equations; Modulation and Coding; Atmospheric and Rain Attenuation; Frequency Selection; Link Budgets
- 13.4 Sizing the Communications Payload
- 13.5 Special Topics
Multiple Access: Sharing Communication Links; Payloads with Onboard Processing; Antijam Techniques; Security; Diversity Techniques; Optical Links

A *communications architecture* is the arrangement, or configuration, of satellites and ground stations in a space system, and the network of communication links that transfers information between them. This chapter discusses this arrangement of links, their operation, and their effect on system design. More detailed information on satellite communications is available in Morgan and Gordon [1989] and Sklar [1988].

Table 13-1 lists the steps required to specify the communications architecture. The first step is to define the mission objectives and requirements in enough detail to evaluate and compare alternative architectures. Section 13.1 describes alternative configurations and the criteria used in their selection.

Communication Modeling

Communication Modeling

Maximum communication bit rate:

$$R_{\max} = B \log_2(1 + C/N)$$

R_{max}: channel bits/sec

B: channel bandwidth

C: received signal power

N: received noise power

Communication Modeling

Maximum communication bit rate:

$$R_{\max} = B \log_2(1 + C/N)$$

R_{max}: channel bits/sec

B: channel bandwidth

C: received signal power

N: received noise power

Received signal power C:

$$C = P L_l G_t \left(\frac{\lambda}{4\pi S} \right)^2 L_a G_r$$

P: transmitted power

L_l: TX to ANT line loss factor

G_t: transmitter gain

λ: channel center freq. wavelength

S: separation vector magnitude

L_a: atmosphere loss factor

G_r: receiver gain

Communication Modeling

Maximum communication bit rate:

$$R_{\max} = B \log_2(1 + C/N)$$

R_{max}: channel bits/sec

B: channel bandwidth

C: received signal power

N: received noise power

Received noise power N:

$$N = kT_sB$$

k: Boltzmann constant

T_s: receiver system noise temperature

B: channel bandwidth

Received signal power C:

$$C = PL_l G_t \left(\frac{\lambda}{4\pi S} \right)^2 L_a G_r$$

P: transmitted power

L_l: TX to ANT line loss factor

G_t: transmitter gain

λ: channel center freq. wavelength

S: separation vector magnitude

L_a: atmosphere loss factor

G_r: receiver gain

Communication Modeling

Maximum communication bit rate:

$$R_{\max} = B \log_2(1 + C/N)$$

R_{max}: channel bits/sec

B: channel bandwidth

C: received signal power

N: received noise power

Received signal power C:

$$C = PL_l G_t \left(\frac{\lambda}{4\pi S} \right)^2 L_a G_r$$

P: transmitted power

L_l: TX to ANT line loss factor

G_t: transmitter gain

λ : channel center freq. wavelength

S: separation vector magnitude

L_a: atmosphere loss factor

G_r: receiver gain

Received noise power N:

$$N = kT_s B$$

k: Boltzmann constant

T_s: receiver system noise temperature

B: channel bandwidth

Receiver system noise temperature T_s:

$$T_s = T_{\text{ant}} + \frac{1 - L_r}{L_r} T_0 + \frac{T_r}{L_r}$$

T_{ant}: RX antenna and environment factors

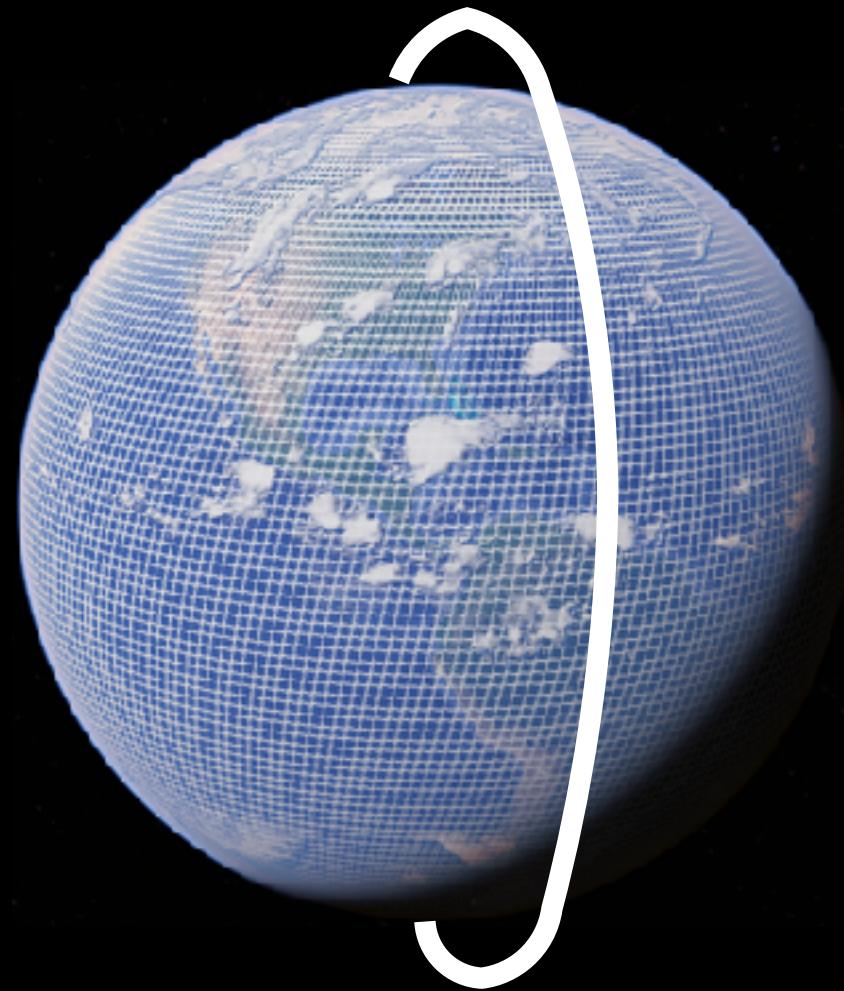
L_r: ANT to RX line loss

T₀: 290 K

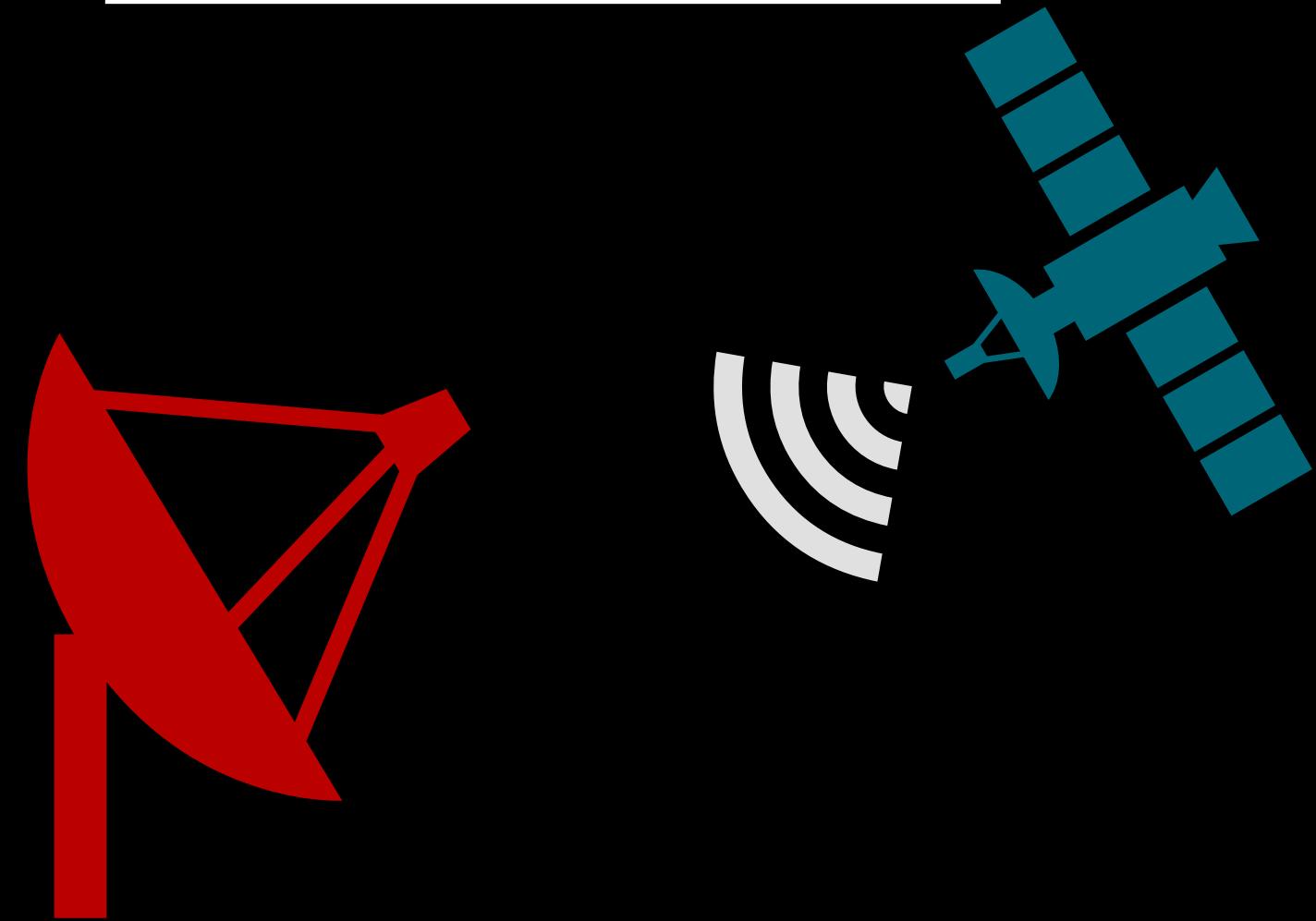
T_r: receiver noise temperature

Components of Space Simulation

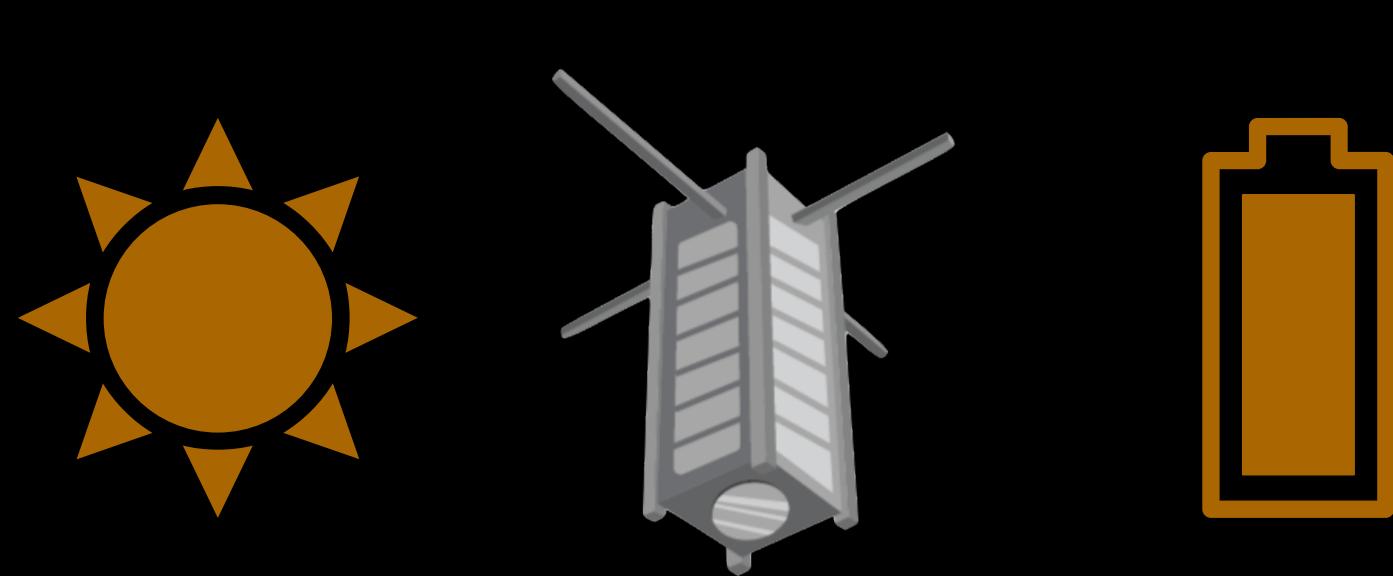
Orbital Mechanics



Communication

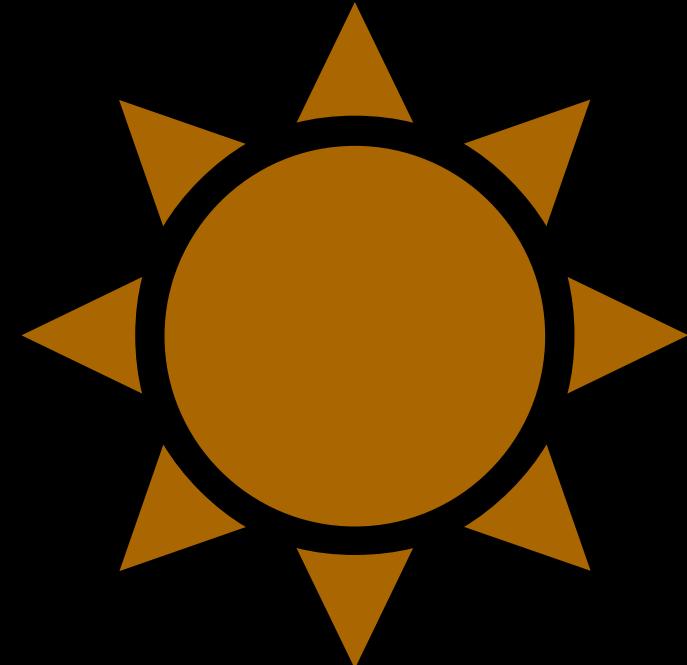


Energy and Power



Satellite Energy Systems

Energy Source

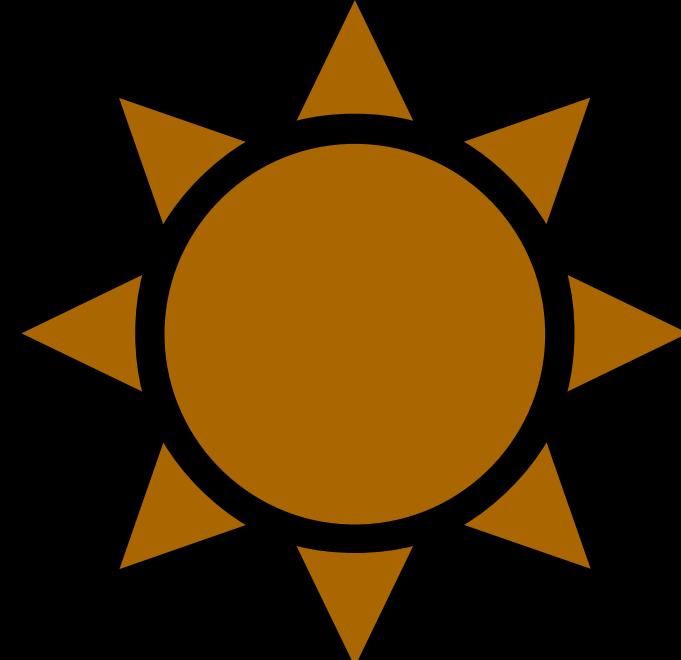


Irradiance

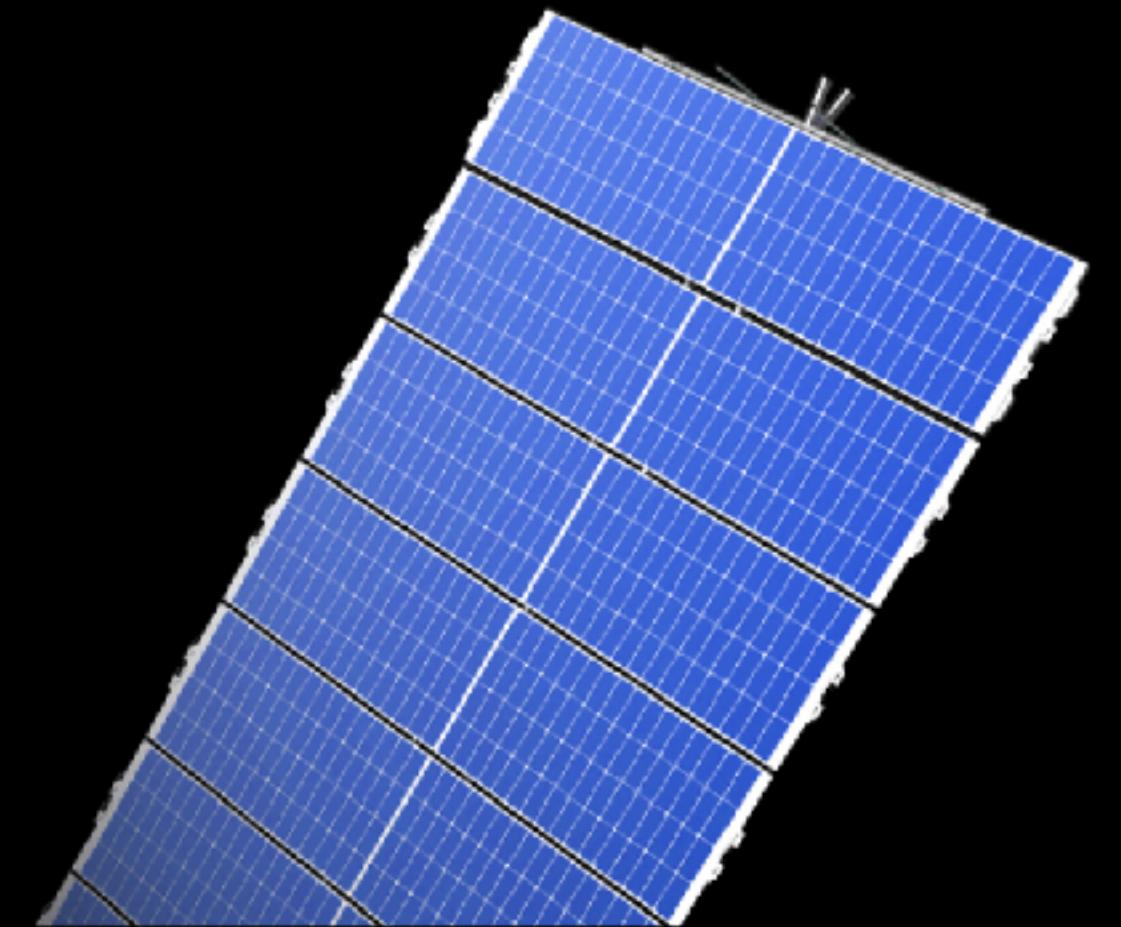
LEO: 1366 W/m²

Satellite Energy Systems

Energy Source



Energy Harvester



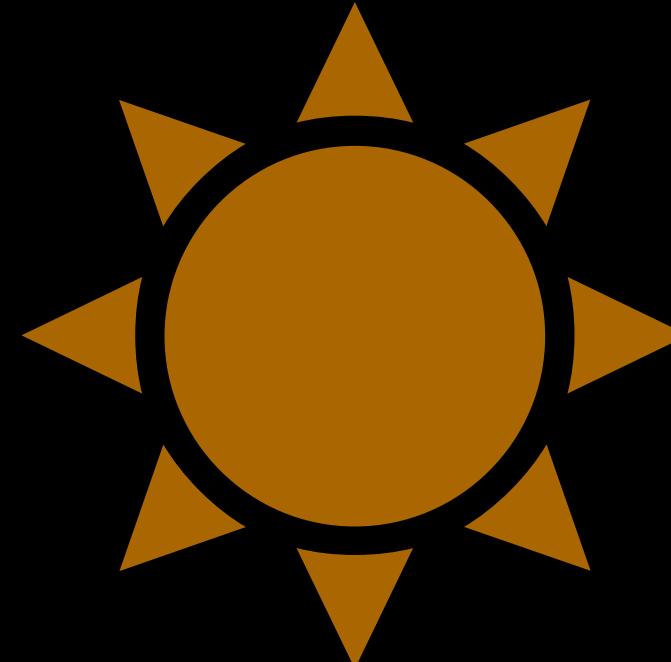
Irradiance

LEO: 1366 W/m²

Surface Area
Efficiency

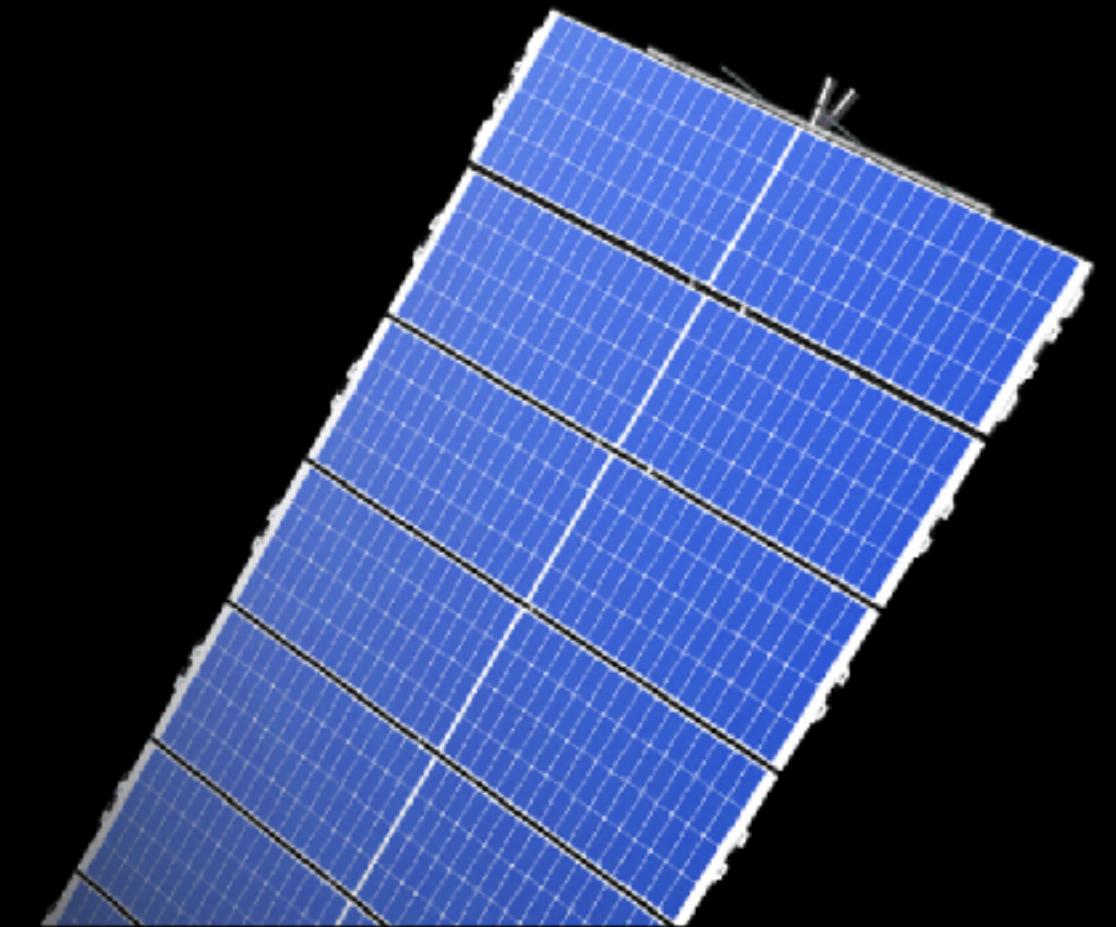
Satellite Energy Systems

Energy Source



Irradiance
LEO: 1366 W/m²

Energy Harvester



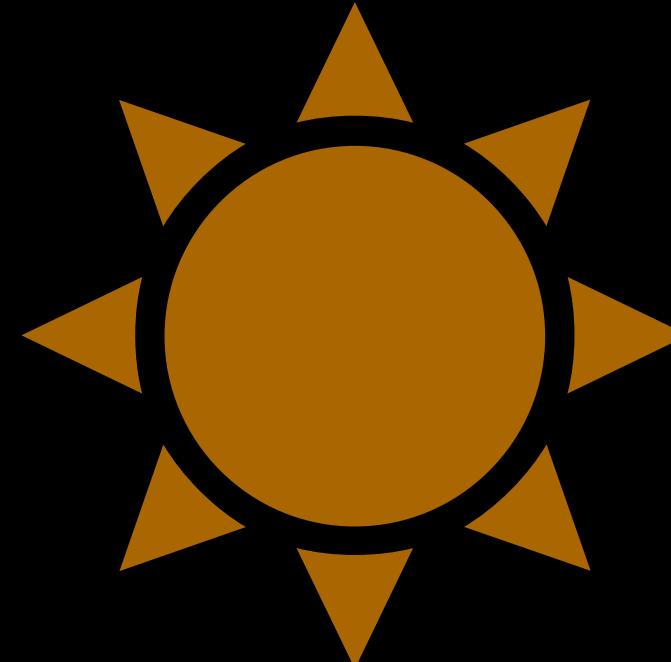
Surface Area
Efficiency

Energy Storage



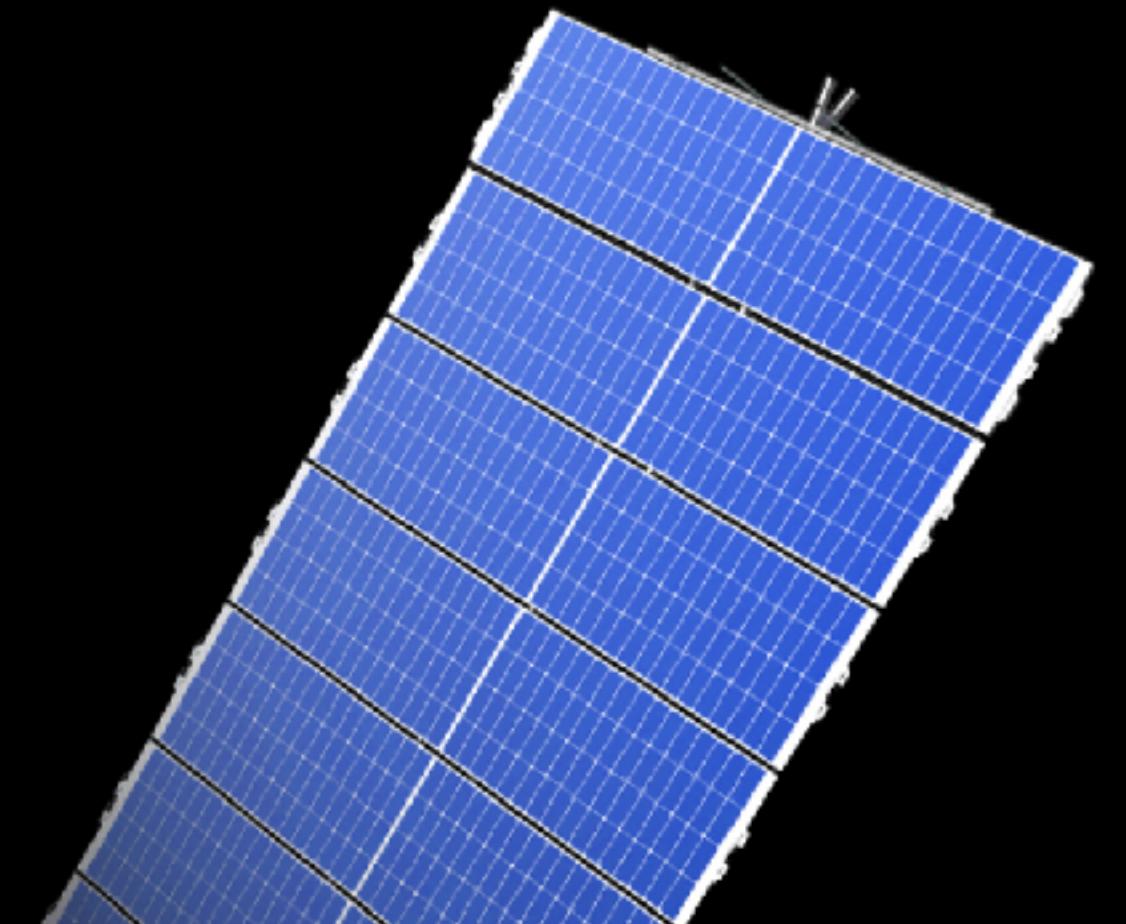
Satellite Energy Systems

Energy Source



Irradiance
LEO: 1366 W/m²

Energy Harvester

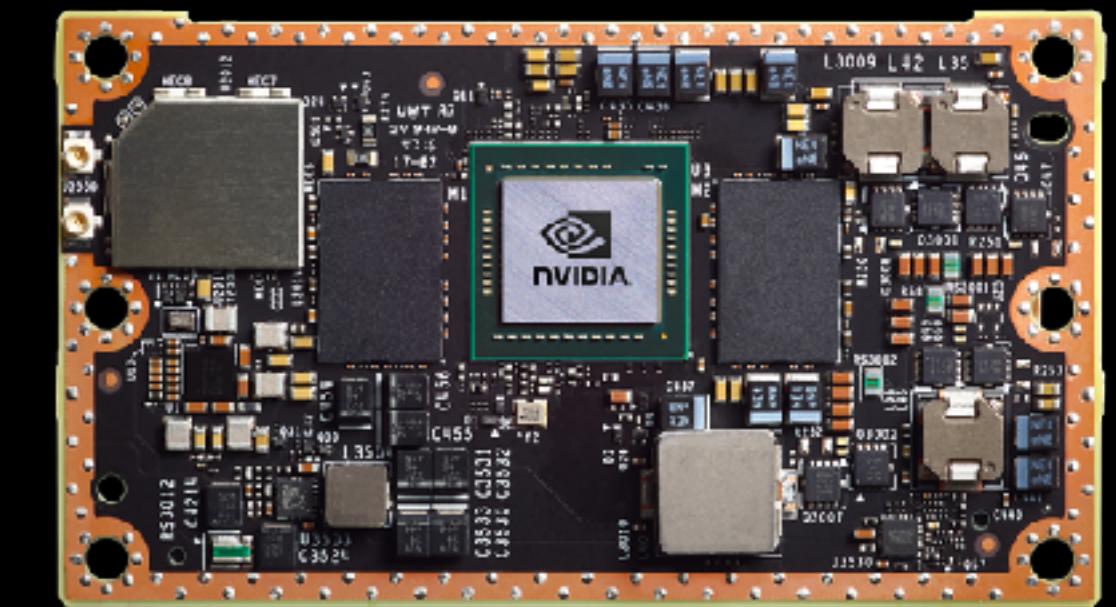


Surface Area
Efficiency

Energy Storage

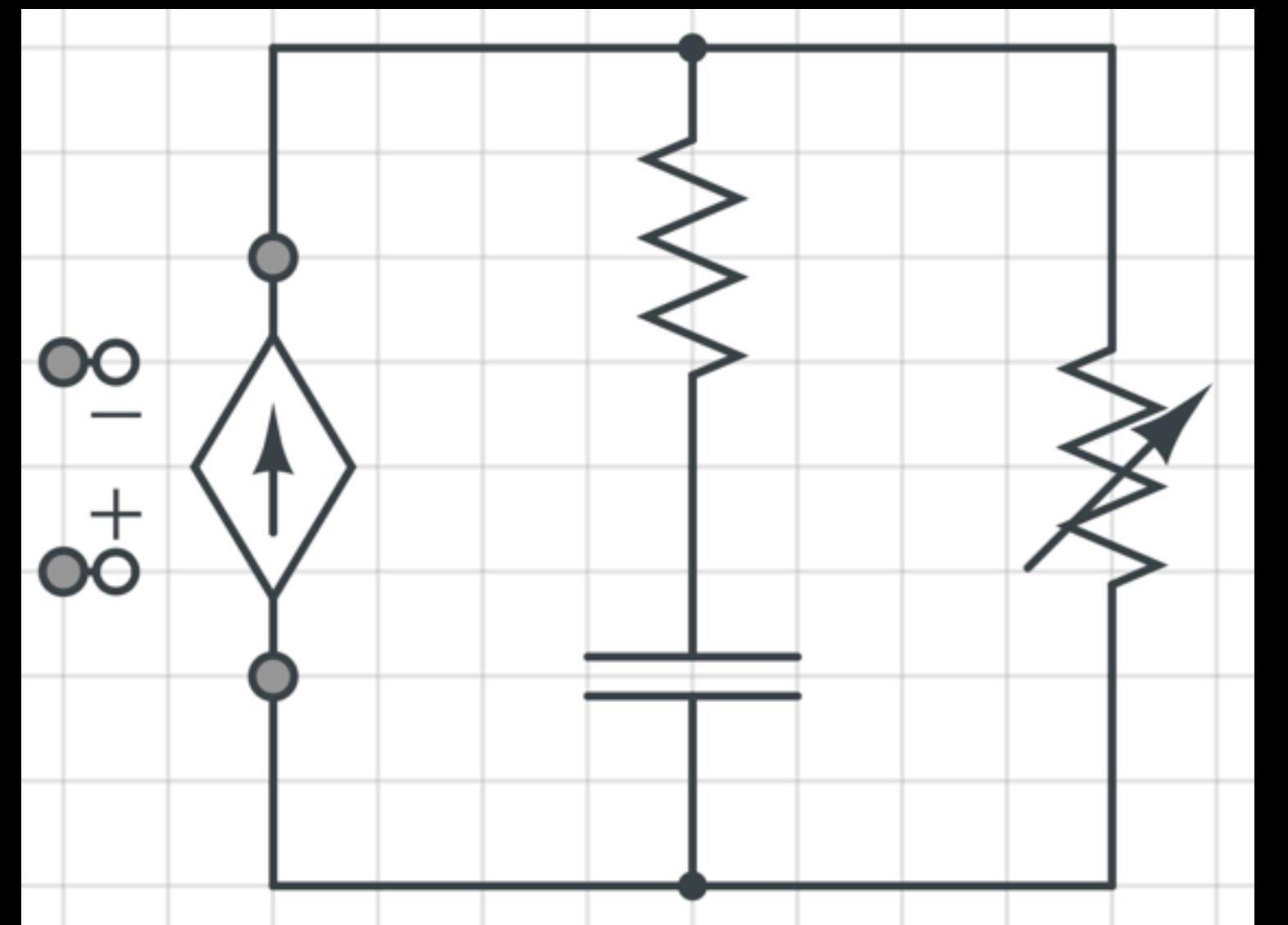


Energy Consumers



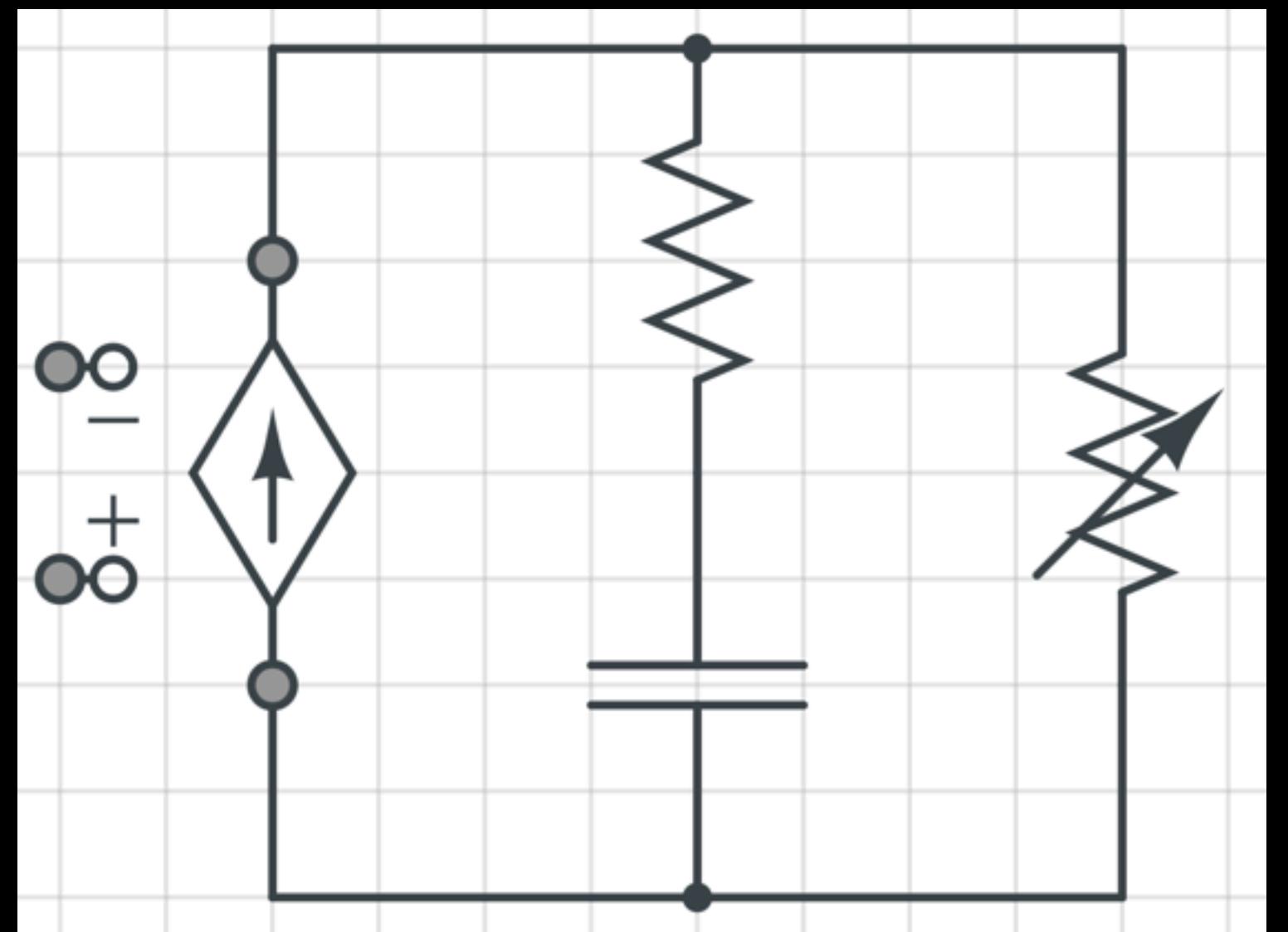
Energy Modeling

Energy Modeling



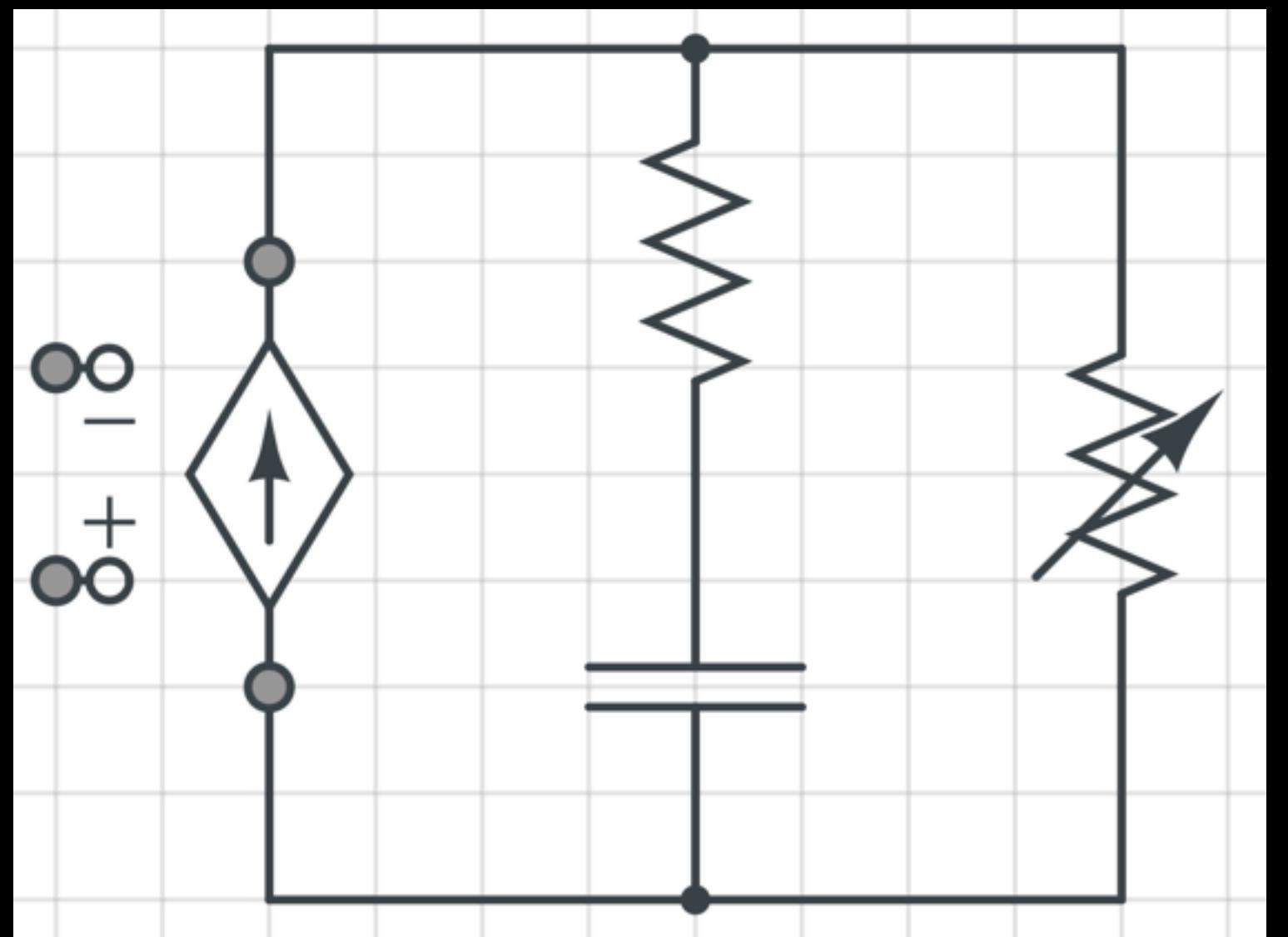
Energy Modeling

Energy harvester:
Solar cell



Energy Modeling

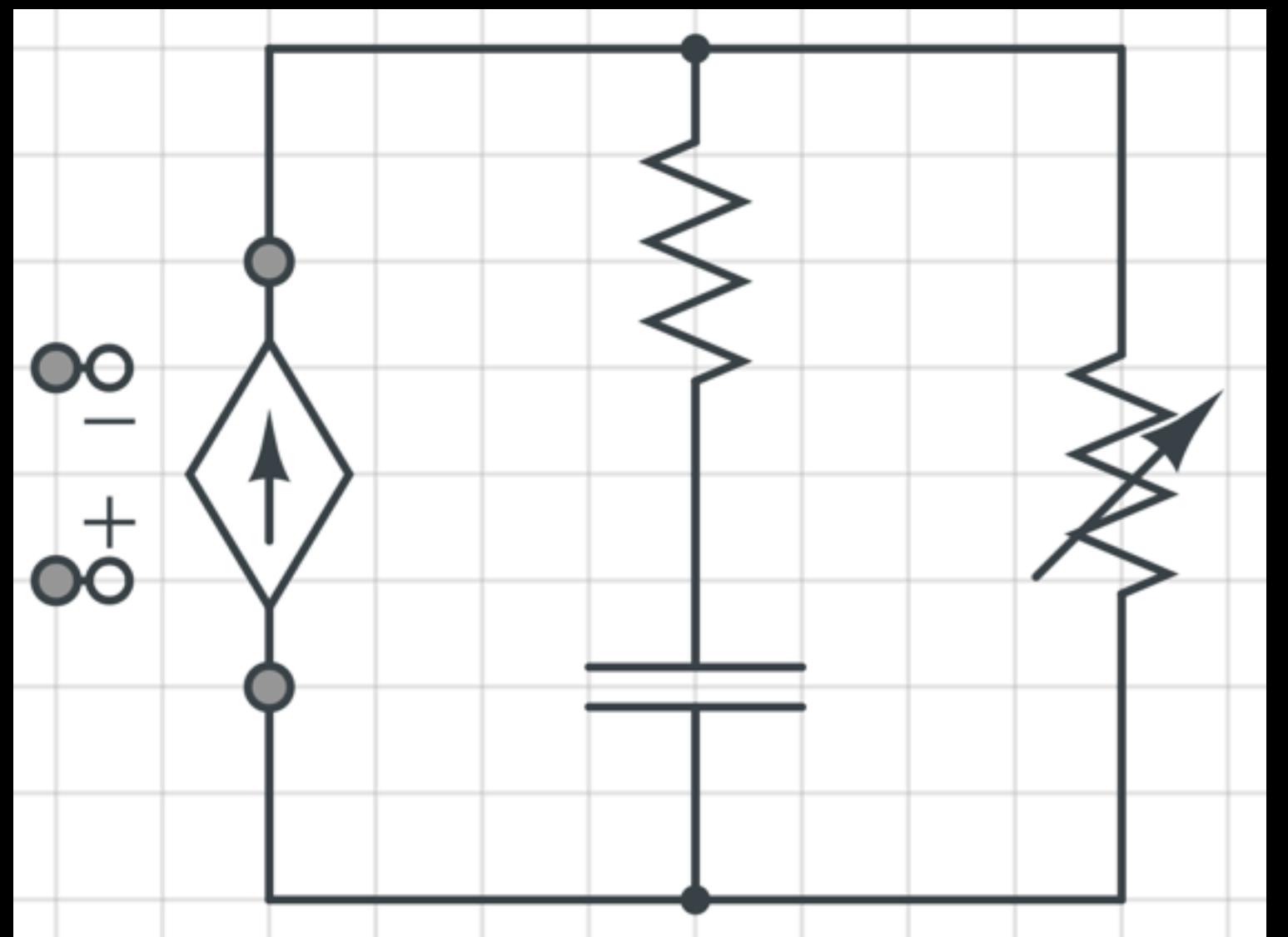
Energy harvester:
Solar cell



Energy storage:
Energy buffer with
Equivalent series resistance

Energy Modeling

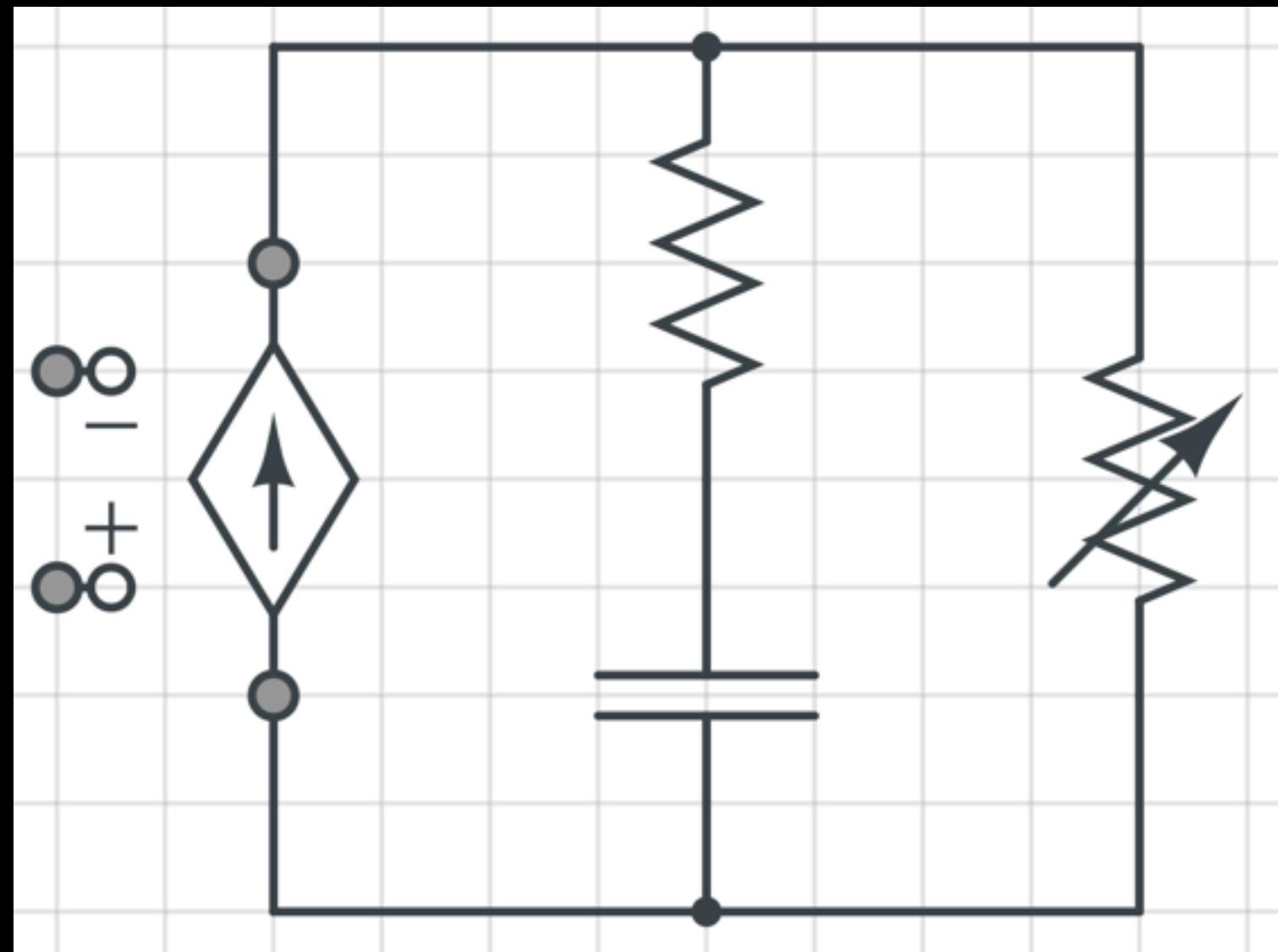
Energy harvester: Energy consumers:
Solar cell Variable resistor



Energy storage:
Energy buffer with
Equivalent series resistance

Energy Modeling

Energy harvester: Energy consumers:
Solar cell Variable resistor



Energy storage:
Energy buffer with
Equivalent series resistance

Solar Array:
E: incident irradiance W/m²
A: solar array surface area
e: solar array efficiency
V_{oc}: open circuit voltage
I_{sc}: short circuit current

IV Curve

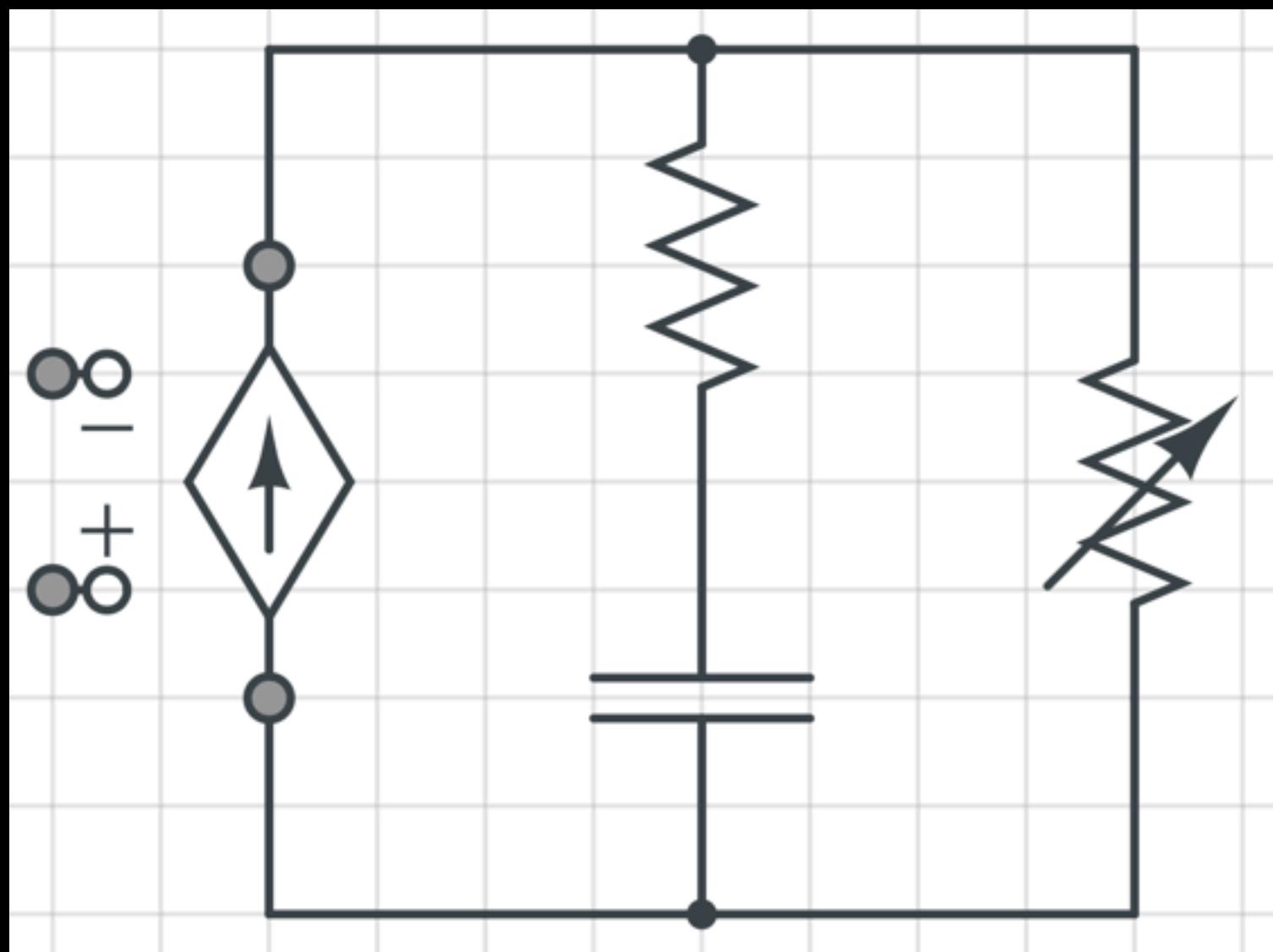
V_{oc}

I_{sc}

$$I_{SC} = E \cdot A \cdot e / V_{OC}$$

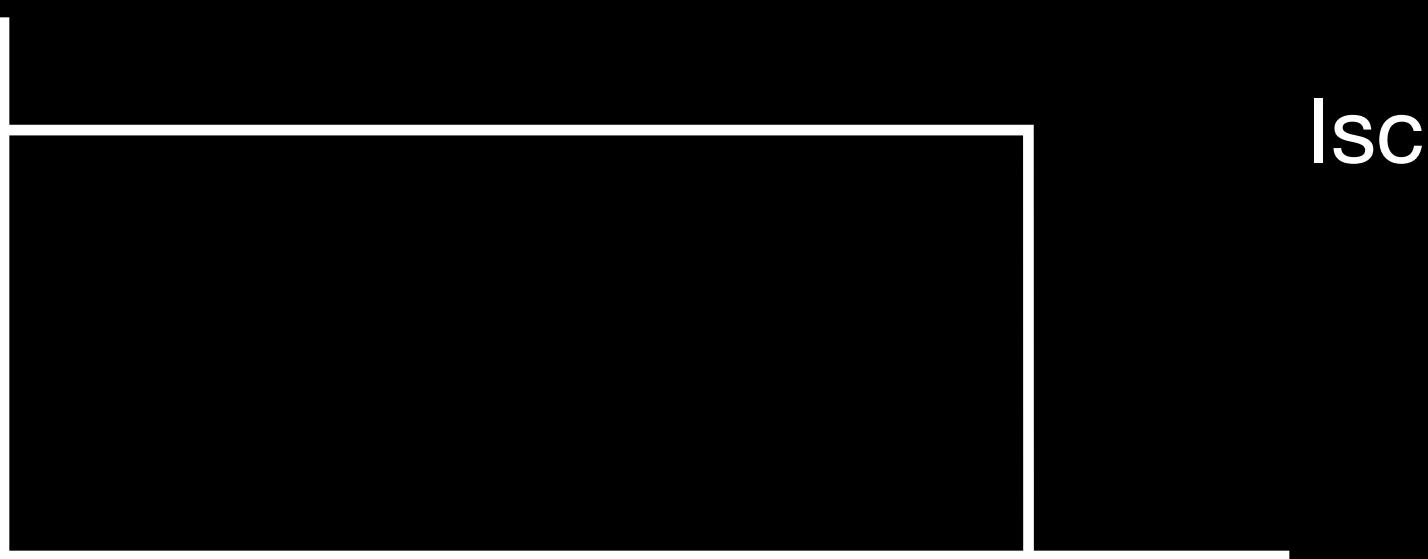
Energy Modeling

Energy harvester: Energy consumers:
Solar cell Variable resistor



Energy storage:
Energy buffer with
Equivalent series resistance

Solar Array:
E: incident irradiance W/m²
A: solar array surface area
e: solar array efficiency
V_{oc}: open circuit voltage
I_{sc}: short circuit current



$$I_{SC} = E \cdot A \cdot e / V_{OC}$$

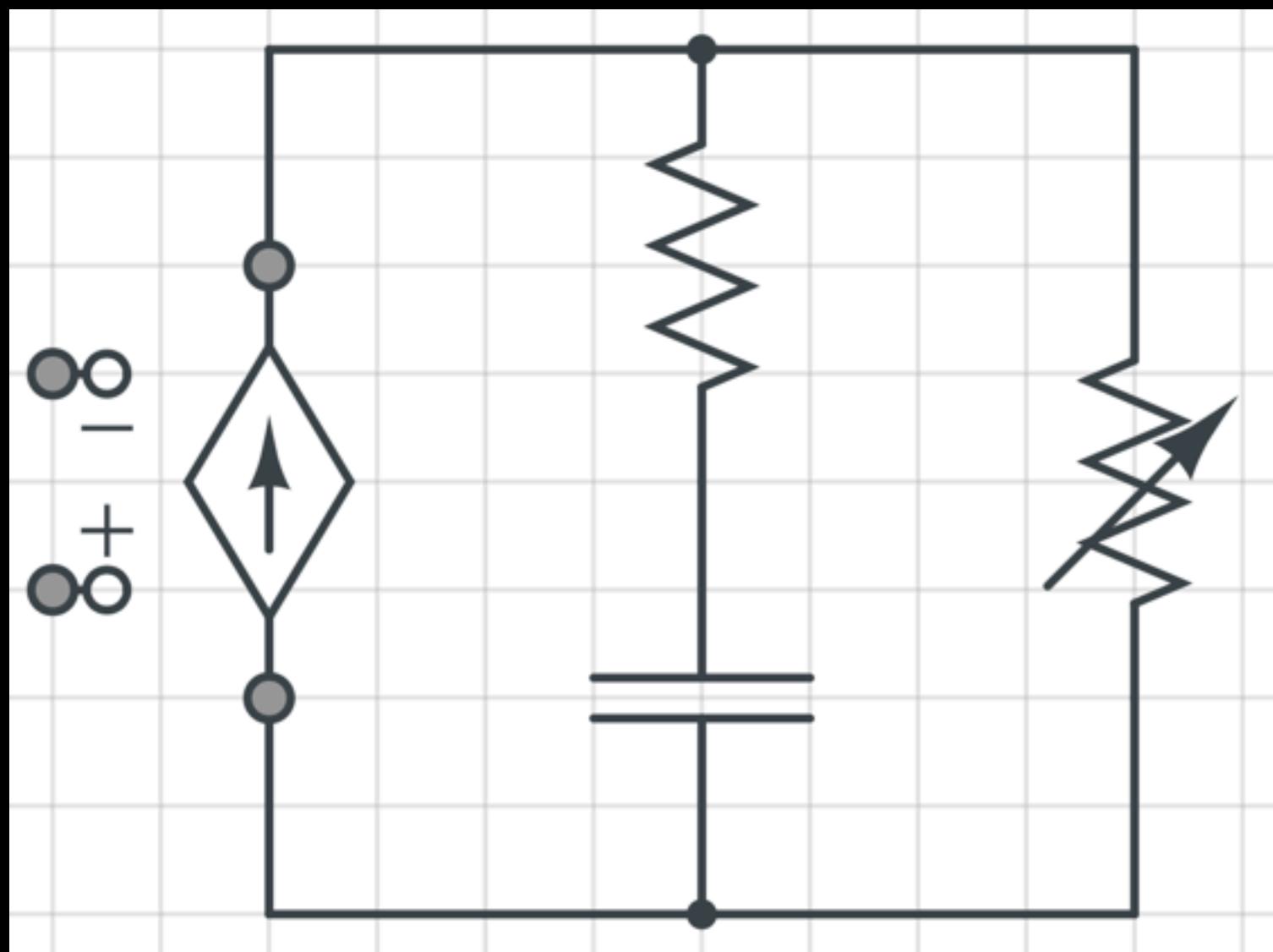
Node Voltage v(t):

$$2v(t) = \frac{q(t)}{C} + I_S R_{ESR} + \sqrt{\left(\frac{q(t)}{C} + I_S R_{ESR}\right)^2 - 4P_{MODE}R_{ESR}}$$

q(t): buffer charge
C: buffer capacitance
I_S: solar array current
R_{ESR}: capacitor ESR
P_{MODE}: peripheral power draw

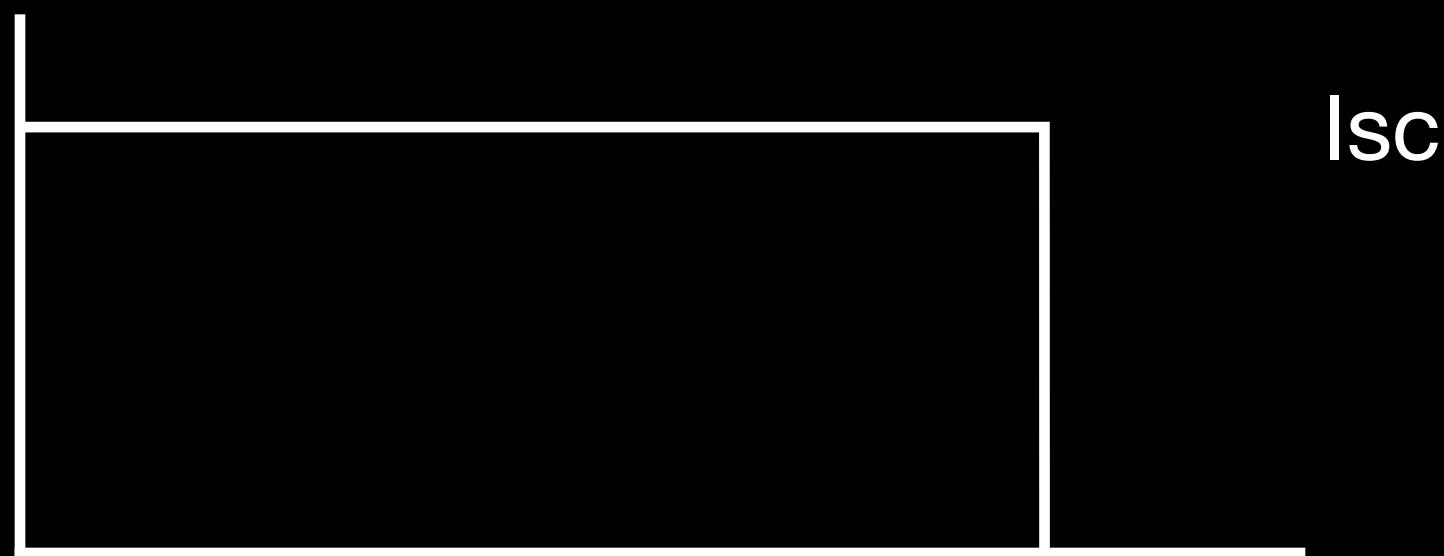
Energy Modeling

Energy harvester: Energy consumers:
Solar cell Variable resistor



Energy storage:
Energy buffer with
Equivalent series resistance

Solar Array:
E: incident irradiance W/m²
A: solar array surface area
e: solar array efficiency
 V_{oc} : open circuit voltage
 I_{sc} : short circuit current



$$I_{SC} = E \cdot A \cdot e / V_{OC}$$

Node Voltage $v(t)$:

$$2v(t) = \frac{q(t)}{C} + I_S R_{ESR} + \sqrt{\left(\frac{q(t)}{C} + I_S R_{ESR}\right)^2 - 4P_{MODE}R_{ESR}}$$

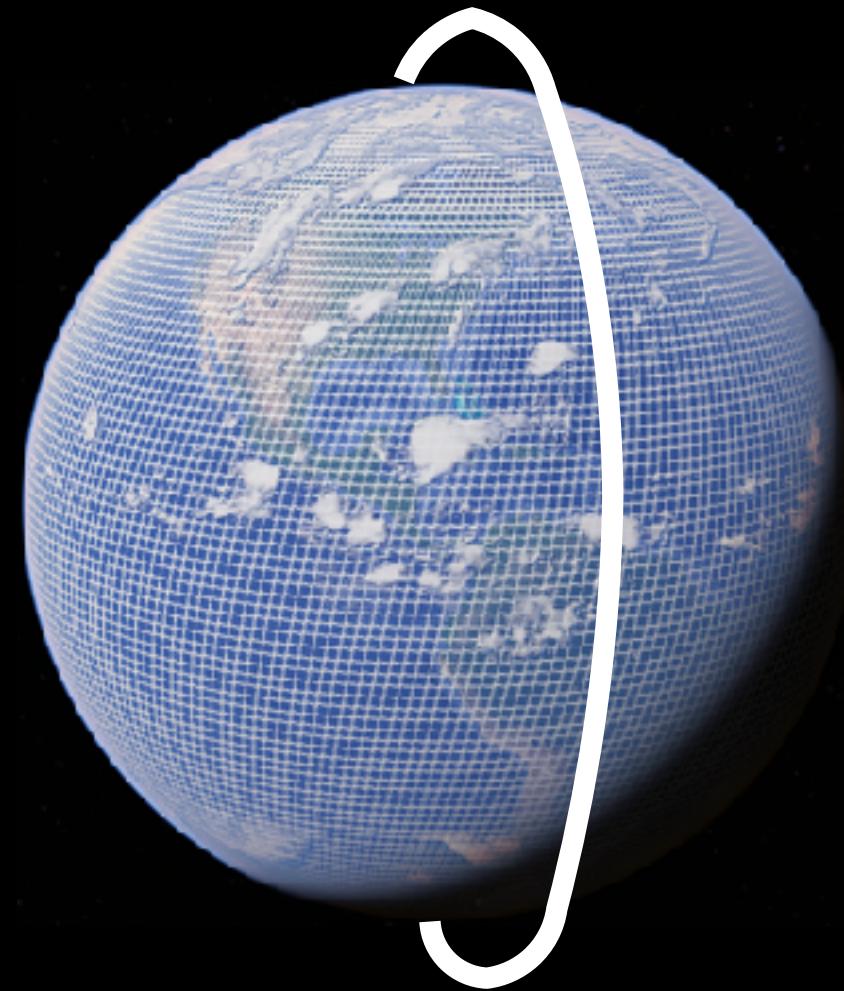
$q(t)$: buffer charge
 C : buffer capacitance
 I_S : solar array current
 R_{ESR} : capacitor ESR
 P_{MODE} : peripheral power draw

Energy consumers:

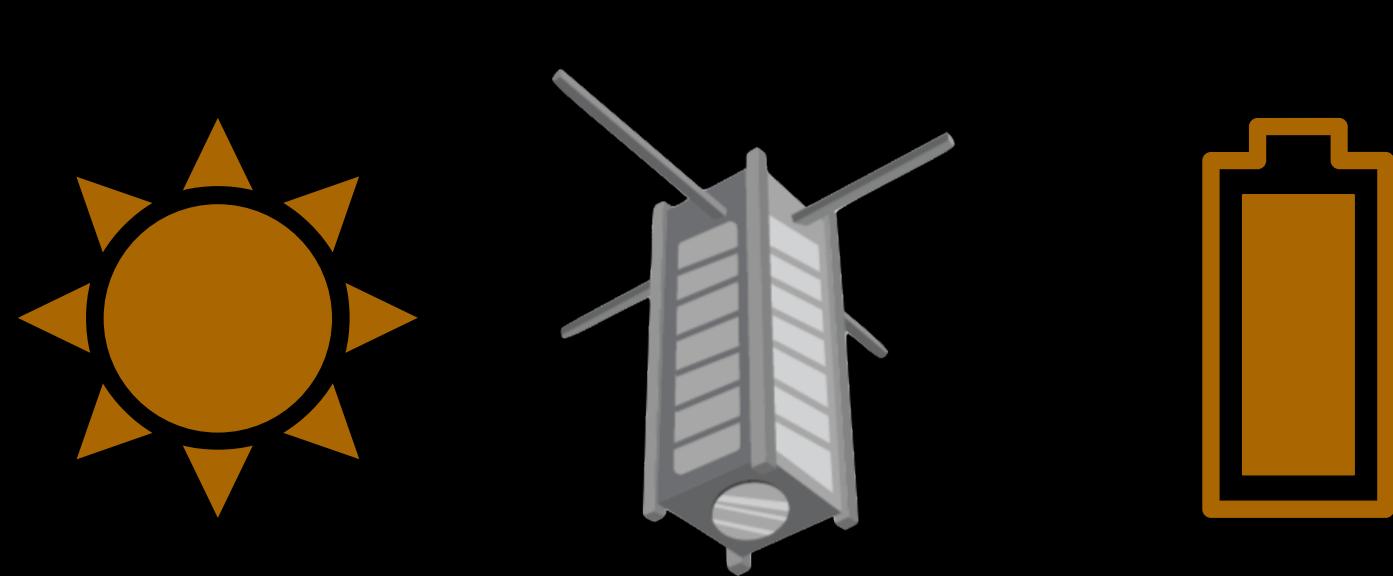
$$I_D = \frac{P_{MODE}}{v(t)}$$

Components of Space Simulation

Orbital Mechanics



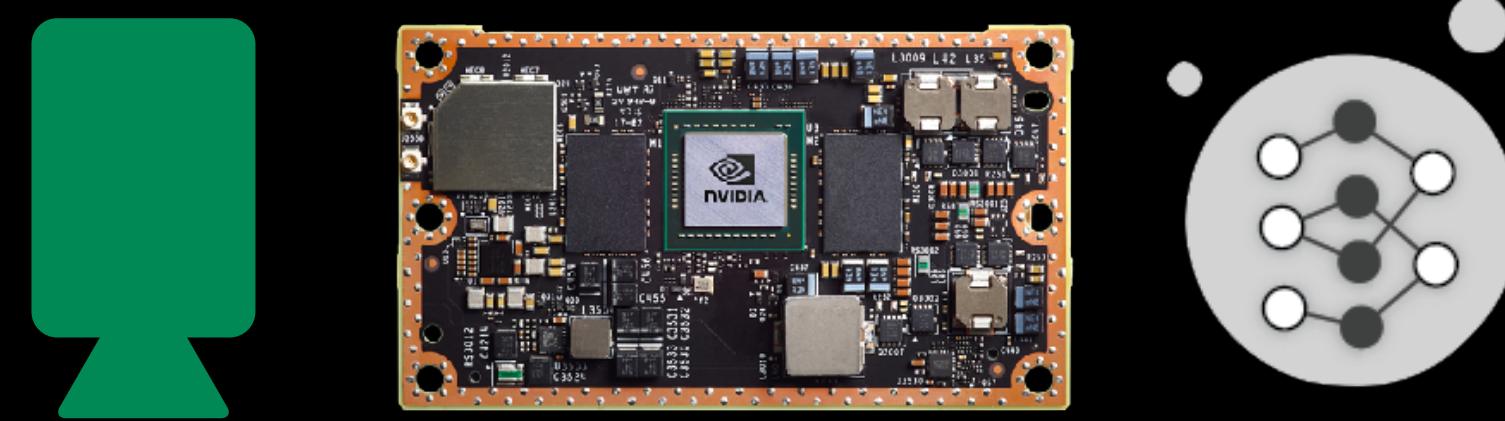
Energy and Power



Communication

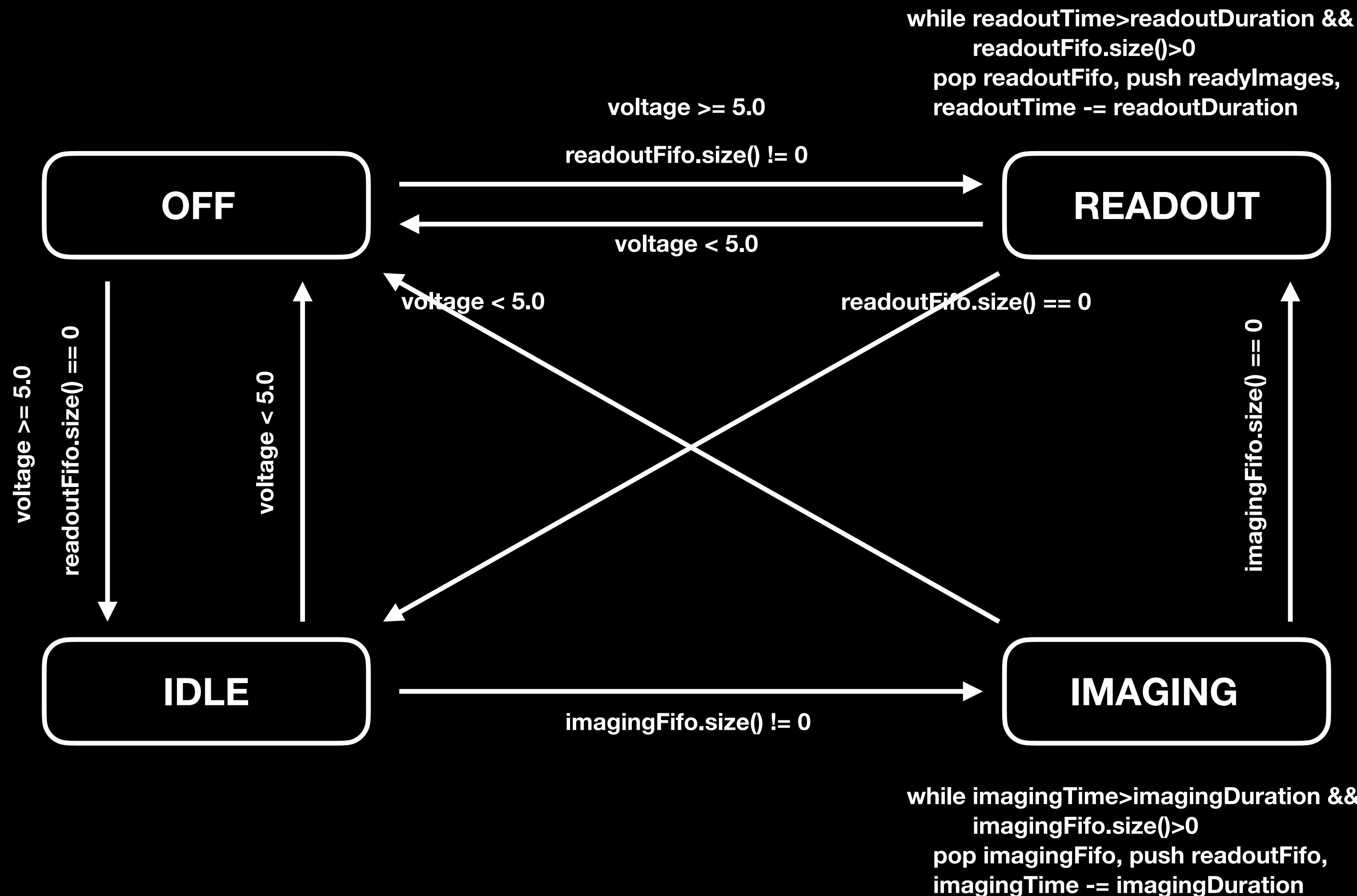


Satellite Subsystems



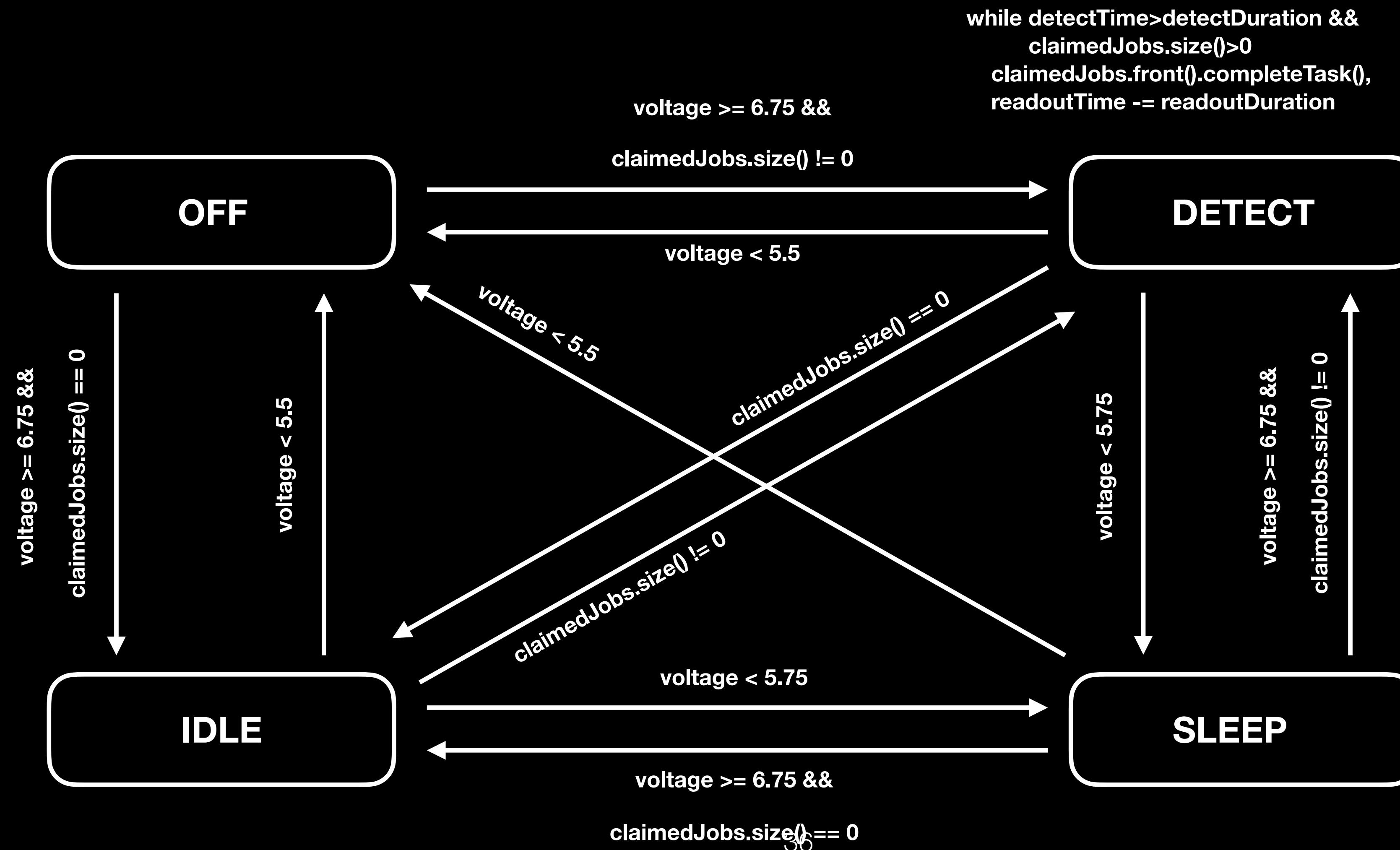
Camera Peripheral Modeling

Imager State Machine



Computing Peripheral Modeling

Processor State Machine



Other Space Simulation Components

Date and Time

```
JD (I, J, K) = K - 32075 + 1461*(I + 4800 + (J - 14)/12)/4  
+ 367*(J - 2 - (J - 14)/12*12)/12 - 3  
*((I + 4900 + (J - 14)/12)/100)/4
```

The authors have yet to discover the algorithm of comparable compactness for converting a Julian Date back to a calendar date. But in preference to leaving the problem undiscussed, the following is offered (presented as a FORTRAN subroutine):

```
SUBROUTINE DATE (JD, I, J, K)  
L = JD + 68569  
N = 4*L/146097  
L = L - (146097*N + 3)/4  
I = 4000*(L + 1)/1461001  
L = L - 1461*I/4 + 31  
J = 80*L/2447  
K = L - 2447*J/80  
L = J/11  
J = J + 2 - 12*L  
I = 100*(N - 49) + I + L  
RETURN  
END
```

HENRY F. FLIEGEL
Georgetown University Observatory
Washington, D.C.
AND
THOMAS C. VAN FLANDERN
U.S. Naval Observatory
Washington, DC 20390

Coordinate Frames

Other Space Simulation Components

Date and Time

```
JD (I, J, K) = K - 32075 + 1461*(I + 4800 + (J - 14)/12)/4  
+ 367*(J - 2 - (J - 14)/12*12)/12 - 3  
*((I + 4900 + (J - 14)/12)/100)/4
```

The authors have yet to discover the algorithm of comparable compactness for converting a Julian Date back to a calendar date. But in preference to leaving the problem undiscussed, the following is offered (presented as a FORTRAN subroutine):

```
SUBROUTINE DATE (JD, I, J, K)  
L = JD + 68569  
N = 4*L/146097  
L = L - (146097*N + 3)/4  
I = 4000*(L + 1)/1461001  
L = L - 1461*I/4 + 31  
J = 80*L/2447  
K = L - 2447*J/80  
L = J/11  
J = J + 2 - 12*L  
I = 100*(N - 49) + I + L  
RETURN  
END
```

HENRY F. FLIEGEL
Georgetown University Observatory
Washington, D.C.
AND
THOMAS C. VAN FLANDERN
U.S. Naval Observatory
Washington, DC 20390

Coordinate Frames

- ECI: Earth-centered inertial
 - (x,y,z) 3D coordinate frame
 - Useful for representing satellite locations
- LLH: Latitude, longitude, and height above the ellipsoid
 - Useful for placing ground stations
- SEZ: South, East, and z-direction
 - Useful for ground station views of satellites above the horizon

Overview

Background and Related Work

Components of Space Simulation

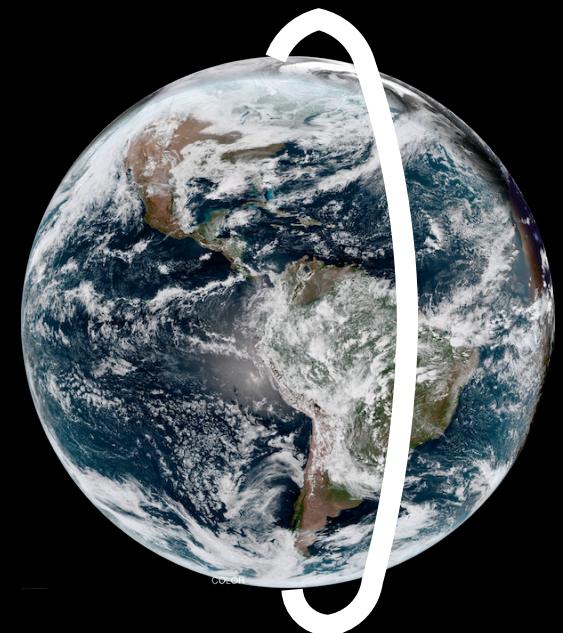
The cote Workflow

Example Programs in cote

The cote Workflow

The cote Workflow

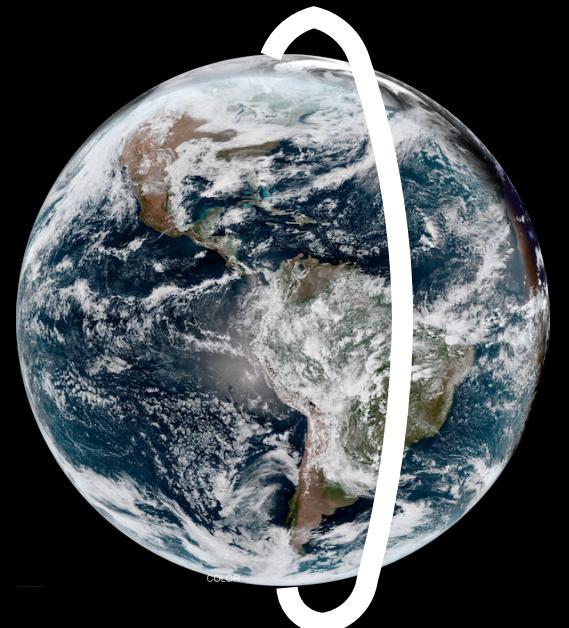
Simulator Inputs



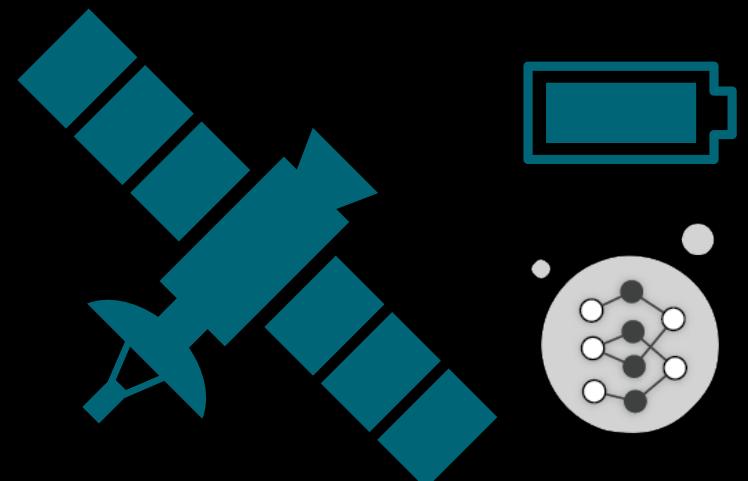
**Orbital Mechanics
Configuration**

The cote Workflow

Simulator Inputs



**Orbital Mechanics
Configuration**

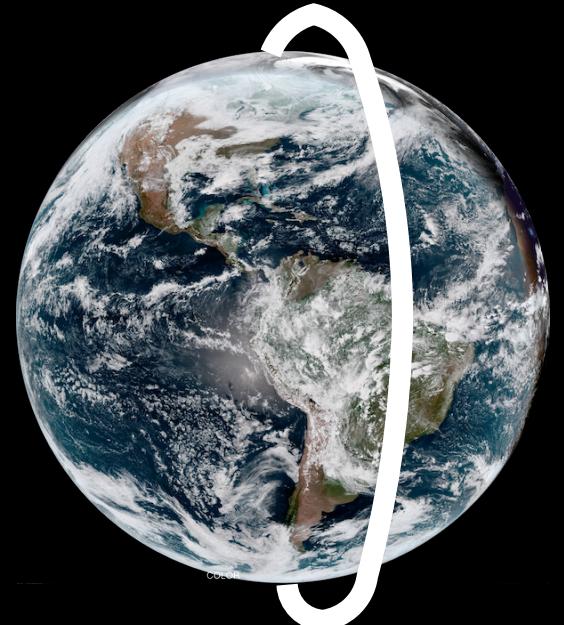


Satellite Configuration:

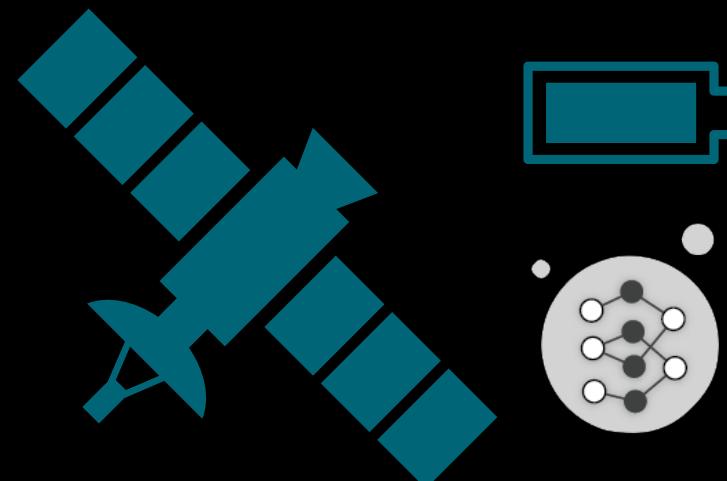
- Energy System
- Onboard Processing
- Other Peripherals

The cote Workflow

Simulator Inputs

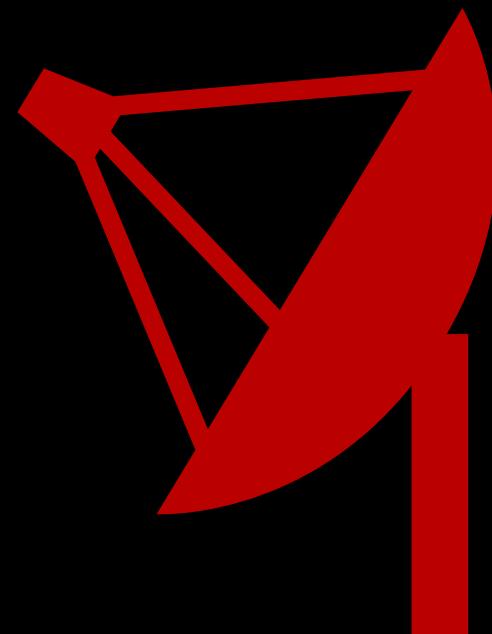


**Orbital Mechanics
Configuration**



Satellite Configuration:

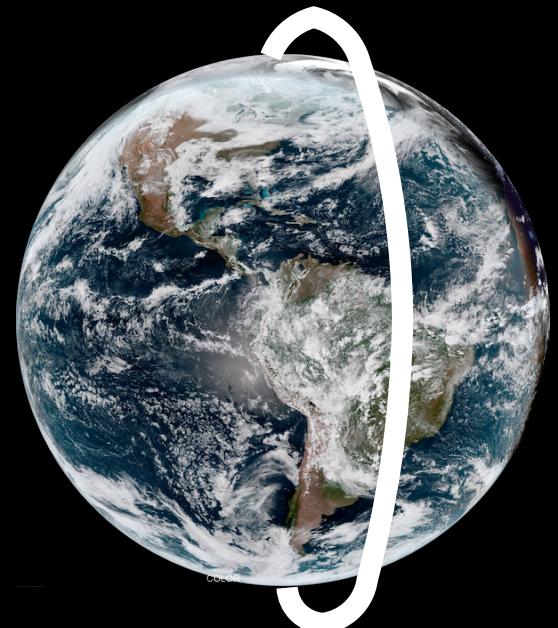
- Energy System
- Onboard Processing
- Other Peripherals



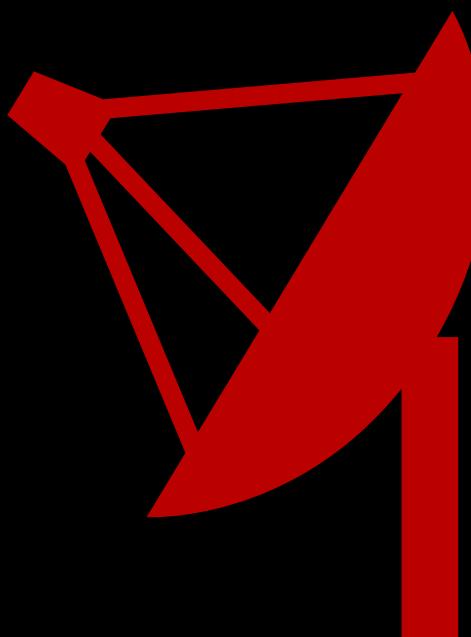
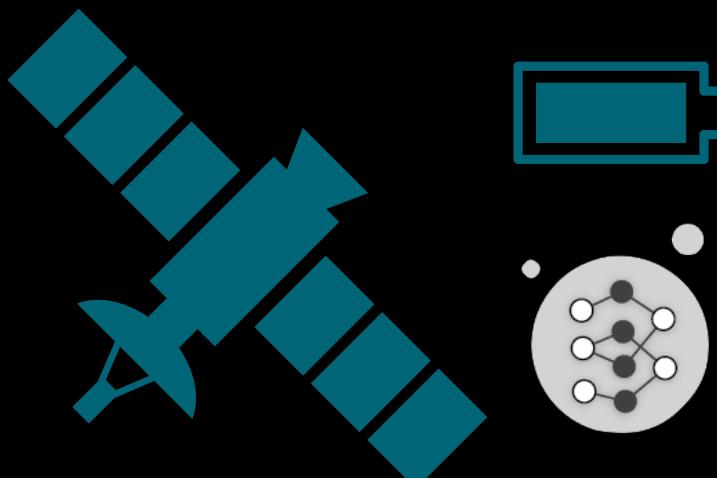
**Communications
Configuration**

The cote Workflow

Simulator Inputs



Orbital Mechanics
Configuration



Satellite Configuration:
• Energy System
• Onboard Processing
• Other Peripherals

Communications
Configuration

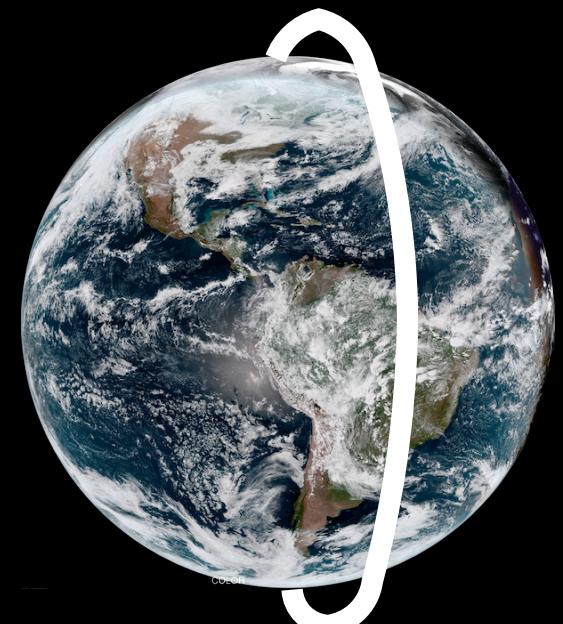
Simulator Outputs

Scenario
Simulation

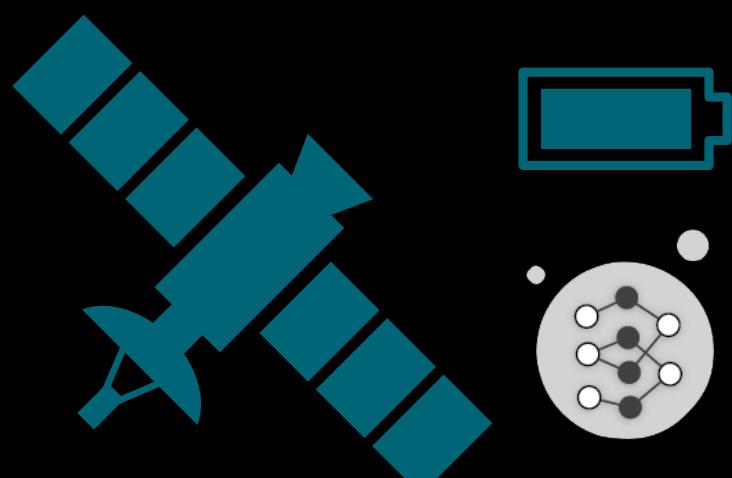


The cote Workflow

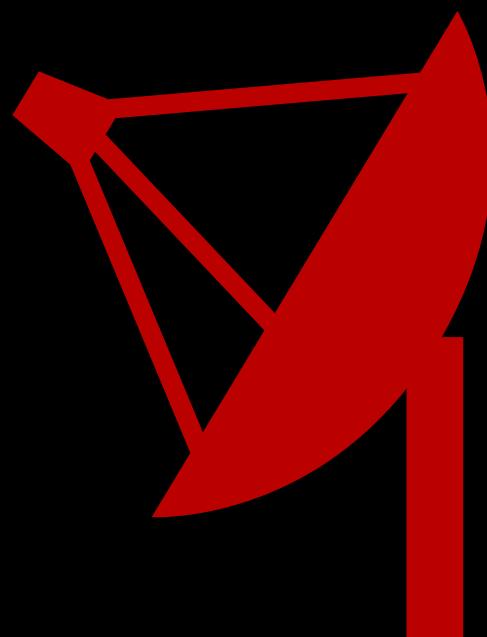
Simulator Inputs



Orbital Mechanics
Configuration



Satellite Configuration:
• Energy System
• Onboard Processing
• Other Peripherals



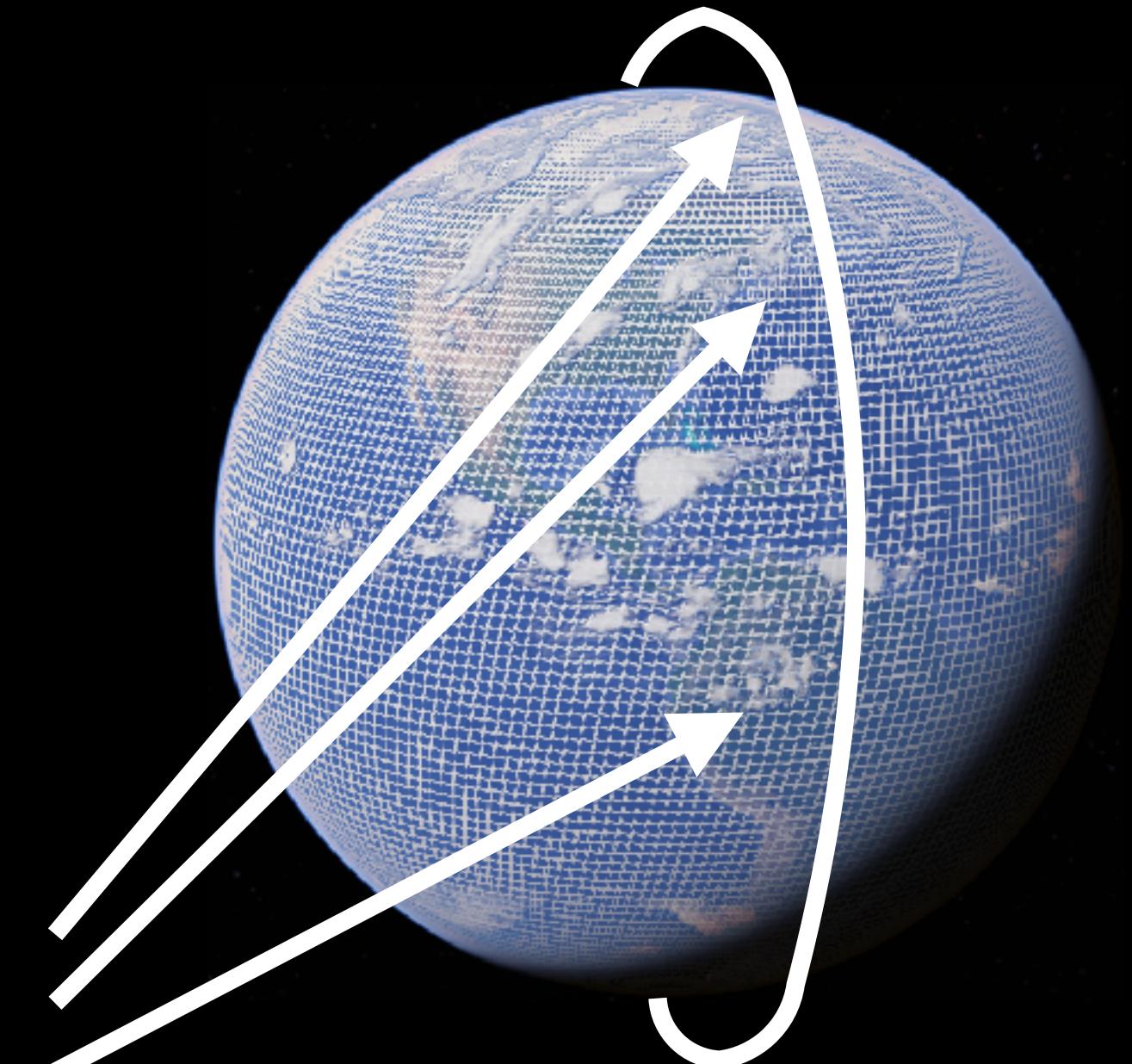
Communications
Configuration

Simulator Outputs

Scenario
Simulation

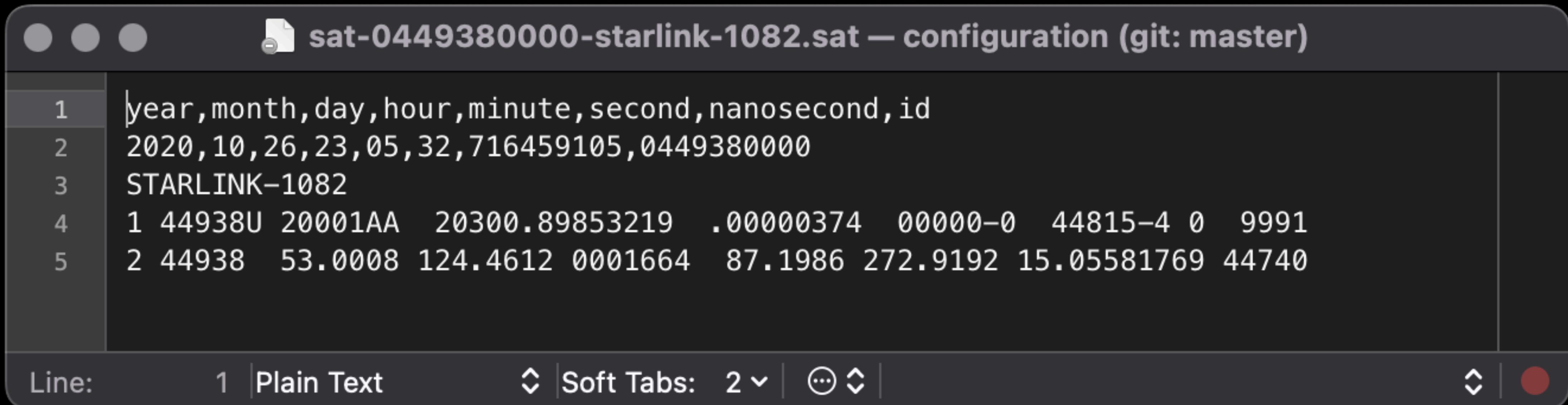


Event Traces



Observation Dates and Times
Satellite Activity Traces
Communication Logs

Example Satellite Config. File



A screenshot of a terminal window titled "sat-0449380000-starlink-1082.sat — configuration (git: master)". The window contains the following text:

```
1 |year,month,day,hour,minute,second,nanosecond,id  
2 |2020,10,26,23,05,32,716459105,0449380000  
3 |STARLINK-1082  
4 |1 44938U 20001AA 20300.89853219 .00000374 00000-0 44815-4 0 9991  
5 |2 44938 53.0008 124.4612 0001664 87.1986 272.9192 15.05581769 44740
```

The terminal interface includes standard navigation and status indicators at the bottom.

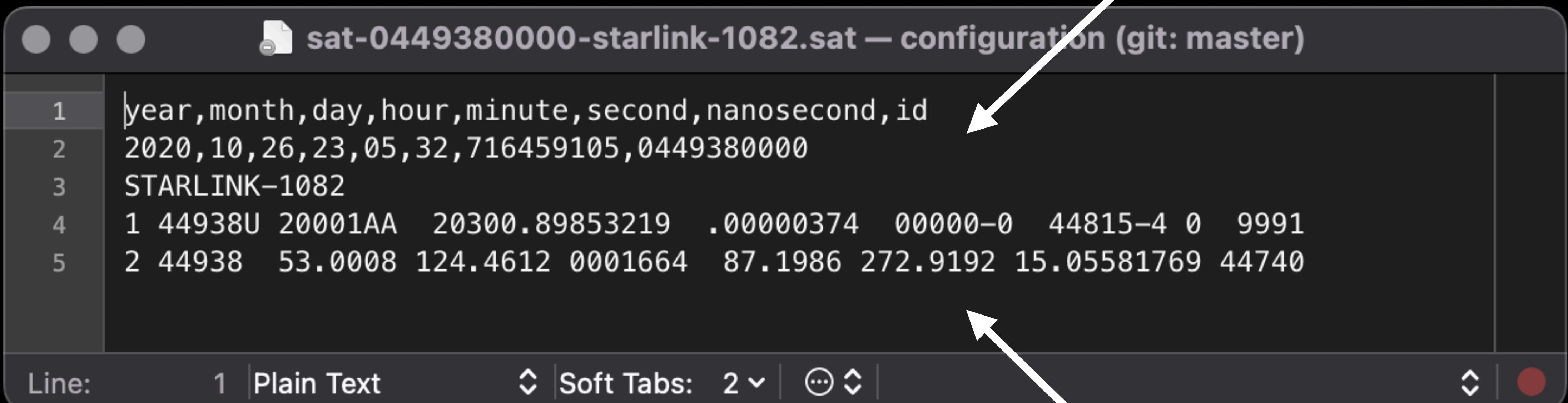
Example Satellite Config. File

```
sat-0449380000-starlink-1082.sat — configuration (git: master)

1 year,month,day,hour,minute,second,nanosecond,id
2 2020,10,26,23,05,32,716459105,0449380000
3 STARLINK-1082
4 1 44938U 20001AA 20300.89853219 .00000374 00000-0 44815-4 0 9991
5 2 44938 53.0008 124.4612 0001664 87.1986 272.9192 15.05581769 44740
```

Satellite Epoch

Example Satellite Config. File



The screenshot shows a terminal window with the following content:

```
1 year,month,day,hour,minute,second,nanosecond,id
2 2020,10,26,23,05,32,716459105,0449380000
3 STARLINK-1082
4 1 44938U 20001AA 20300.89853219 .00000374 00000-0 44815-4 0 9991
5 2 44938 53.0008 124.4612 0001664 87.1986 272.9192 15.05581769 44740
```

Annotations with arrows:

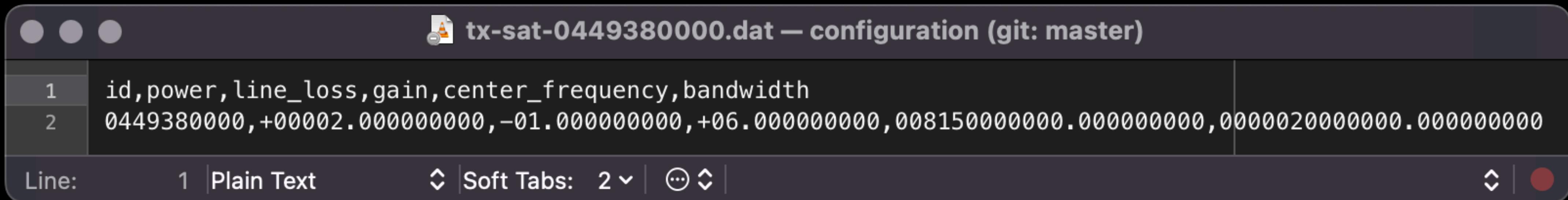
- An arrow points from the text "year,month,day,hour,minute,second,nanosecond,id" to the label "Satellite Epoch".
- An arrow points from the text "1 44938U 20001AA 20300.89853219 .00000374 00000-0 44815-4 0 9991" to the label "Satellite TLE".

Satellite Epoch

Satellite TLE

Example Comms. Config. Files

Example Comms. Config. Files

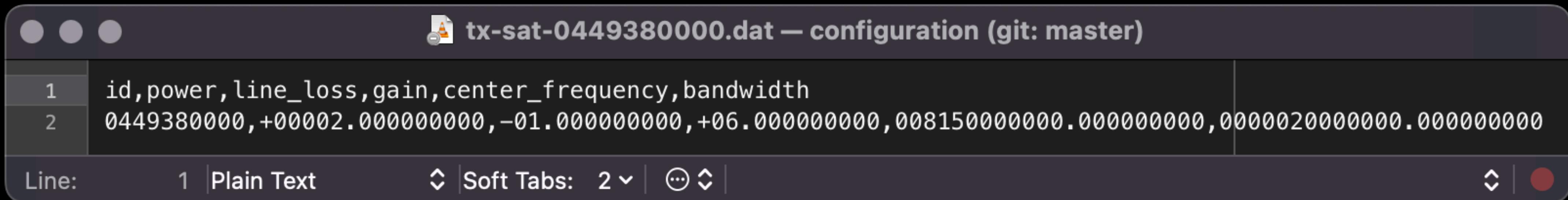


A screenshot of a terminal window titled "tx-sat-0449380000.dat – configuration (git: master)". The window shows two lines of text:

```
1 id,power,line_loss,gain,center_frequency,bandwidth
2 0449380000,+0002.00000000,-01.00000000,+06.00000000,00815000000.00000000,00002000000.00000000
```

The terminal interface includes standard navigation and status bars at the bottom.

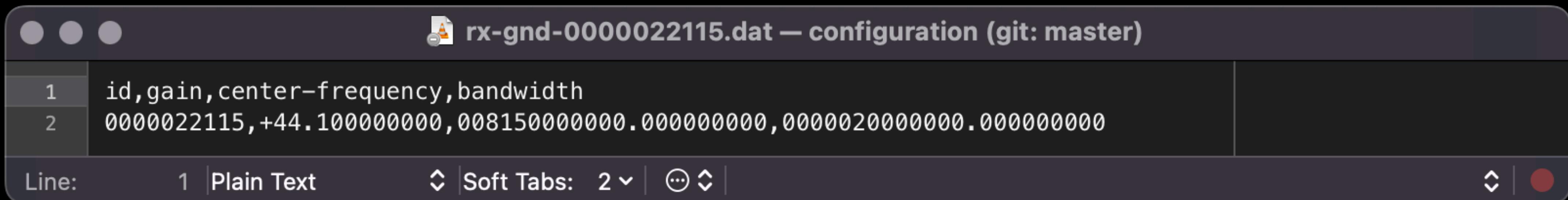
Example Comms. Config. Files



A screenshot of a terminal window titled "tx-sat-0449380000.dat – configuration (git: master)". The window shows two lines of configuration data:

```
1 id,power,line_loss,gain,center_frequency,bandwidth
2 0449380000,+0002.00000000,-01.00000000,+06.00000000,00815000000.00000000,00002000000.00000000
```

The terminal interface includes standard navigation and status bars at the bottom.

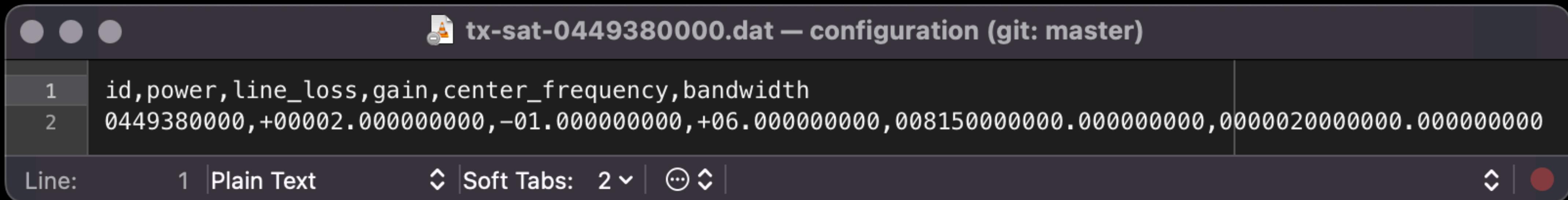


A screenshot of a terminal window titled "rx-gnd-0000022115.dat – configuration (git: master)". The window shows two lines of configuration data:

```
1 id,gain,center-frequency,bandwidth
2 0000022115,+44.10000000,00815000000.00000000,00002000000.00000000
```

The terminal interface includes standard navigation and status bars at the bottom.

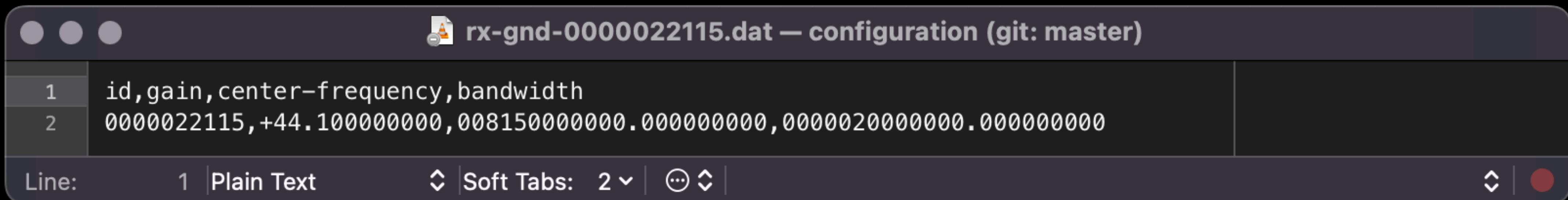
Example Comms. Config. Files



A screenshot of a terminal window titled "tx-sat-0449380000.dat – configuration (git: master)". The window shows two lines of configuration data:

```
1 id,power,line_loss,gain,center_frequency,bandwidth
2 0449380000,+0002.00000000,-01.00000000,+06.00000000,00815000000.00000000,00002000000.00000000
```

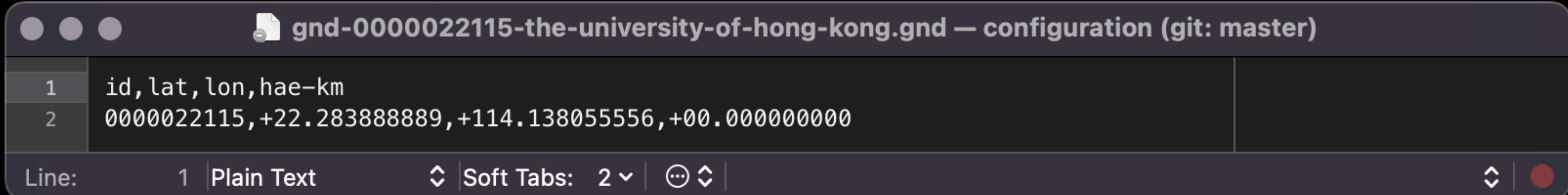
The terminal interface includes standard navigation and status bars at the bottom.



A screenshot of a terminal window titled "rx-gnd-0000022115.dat – configuration (git: master)". The window shows two lines of configuration data:

```
1 id,gain,center-frequency,bandwidth
2 0000022115,+44.10000000,00815000000.00000000,00002000000.00000000
```

The terminal interface includes standard navigation and status bars at the bottom.



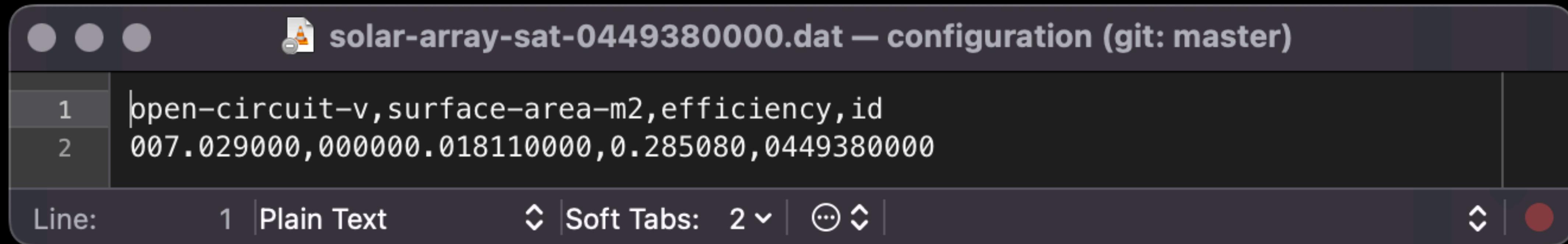
A screenshot of a terminal window titled "gnd-0000022115-the-university-of-hong-kong.gnd – configuration (git: master)". The window shows two lines of configuration data:

```
1 id,lat,lon,hae-km
2 0000022115,+22.283888889,+114.138055556,+00.00000000
```

The terminal interface includes standard navigation and status bars at the bottom.

Example Energy Config. Files

Example Energy Config. Files

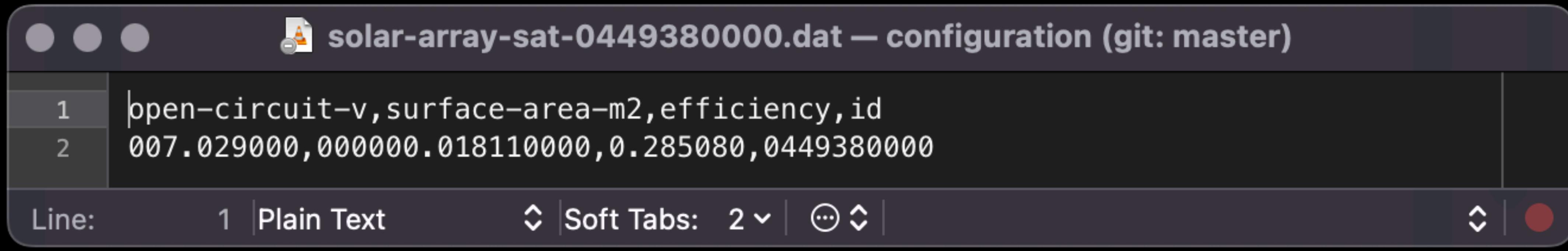


A screenshot of a terminal window titled "solar-array-sat-044938000.dat — configuration (git: master)". The window shows two lines of text:

```
1 open-circuit-v,surface-area-m2,efficiency,id  
2 007.029000,000000.018110000,0.285080,044938000
```

The terminal interface includes a status bar at the bottom with "Line: 1 | Plain Text", "Soft Tabs: 2", and other icons.

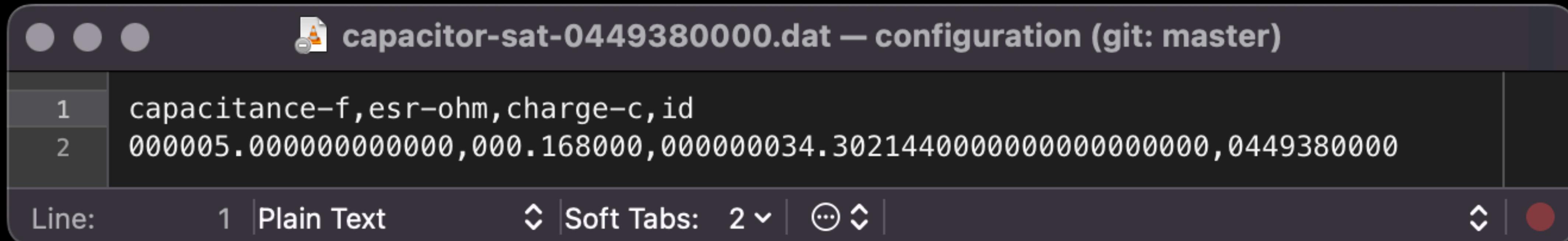
Example Energy Config. Files



A screenshot of a terminal window titled "solar-array-sat-0449380000.dat — configuration (git: master)". The window shows two lines of text:

```
1 open-circuit-v,surface-area-m2,efficiency,id  
2 007.029000,000000.018110000,0.285080,0449380000
```

The terminal interface includes a line number column, a status bar at the bottom with "Line: 1 Plain Text", and various control icons.



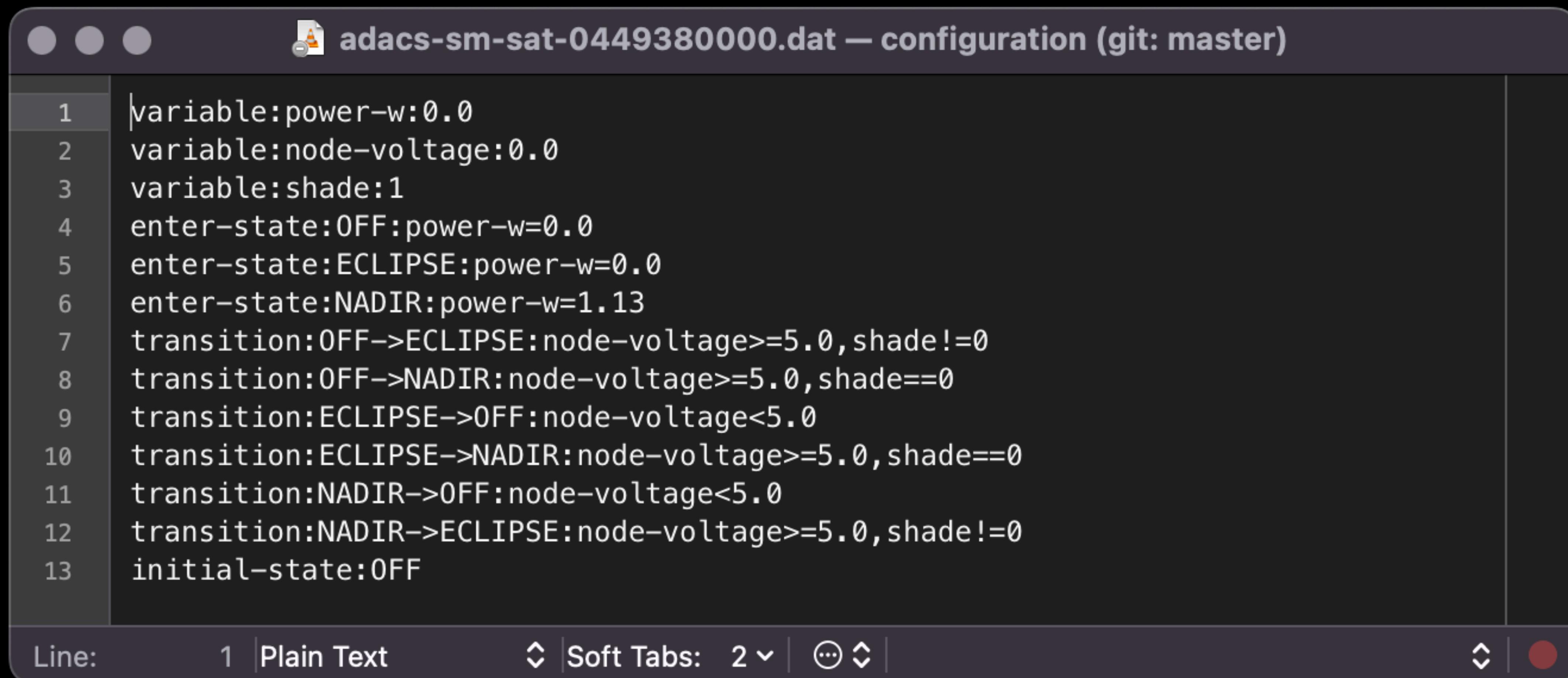
A screenshot of a terminal window titled "capacitor-sat-0449380000.dat — configuration (git: master)". The window shows two lines of text:

```
1 capacitance-f,esr-ohm,charge-c,id  
2 000005.000000000000,000.168000,00000034.3021440000000000000000,0449380000
```

The terminal interface includes a line number column, a status bar at the bottom with "Line: 1 Plain Text", and various control icons.

Example Subsystem Config. Files

Three-state ADACS State Machine



The screenshot shows a terminal window with the title bar "adacs-sm-sat-0449380000.dat – configuration (git: master)". The main area contains a configuration file for a three-state ADACS state machine. The file includes variable definitions, initial states, and transition rules based on node voltage and shade.

```
variable:power-w:0.0
variable:node-voltage:0.0
variable:shade:1
enter-state:OFF:power-w=0.0
enter-state:ECLIPSE:power-w=0.0
enter-state:NADIR:power-w=1.13
transition:OFF->ECLIPSE:node-voltage>=5.0, shade!=0
transition:OFF->NADIR:node-voltage>=5.0, shade==0
transition:ECLIPSE->OFF:node-voltage<5.0
transition:ECLIPSE->NADIR:node-voltage>=5.0, shade==0
transition:NADIR->OFF:node-voltage<5.0
transition:NADIR->ECLIPSE:node-voltage>=5.0, shade!=0
initial-state:OFF
```

At the bottom of the terminal window, there are status indicators: "Line: 1 | Plain Text" and "Soft Tabs: 2". There are also small icons for file operations like copy, paste, and save.

Example Subsystem Config. Files

Imager peripheral:

```
1 constant:imaging-duration-s:0.031260
2 constant:readout-duration-s:0.838860
3 variable:imaging-time-s:0.0
4 variable:readout-time-s:0.0
5 variable:power-w:0.0
6 variable:node-voltage:0.0
7 variable:imaging-task-count:0
8 variable:readout-task-count:0
9 variable:ready-images:0
10 enter-state:OFF:power-w=0.0
11 enter-state:IDLE:power-w=0.0
12 enter-state:IMAGING:power-w=3.5
13 enter-state:READOUT:power-w=2.5
14 exit-state:IMAGING:imaging-time-s=0.0
15 transition:OFF->READOUT:node-voltage>=5.0, readout-task-count!=0
16 transition:OFF->IDLE:node-voltage>=5.0, readout-task-count==0
17 transition:IDLE->OFF:node-voltage<5.0
18 transition:IDLE->IMAGING:node-voltage>=5.0, imaging-task-count!=0
19 transition:IMAGING->OFF:node-voltage<5.0:imaging-task-count=0
20 transition:IMAGING->READOUT:node-voltage>=5.0, imaging-task-count==0
21 transition:READOUT->OFF:node-voltage<5.0
22 transition:READOUT->IDLE:node-voltage>=5.0, readout-task-count==0: readout-time-s=0.0
23 initial-state:OFF
24
```

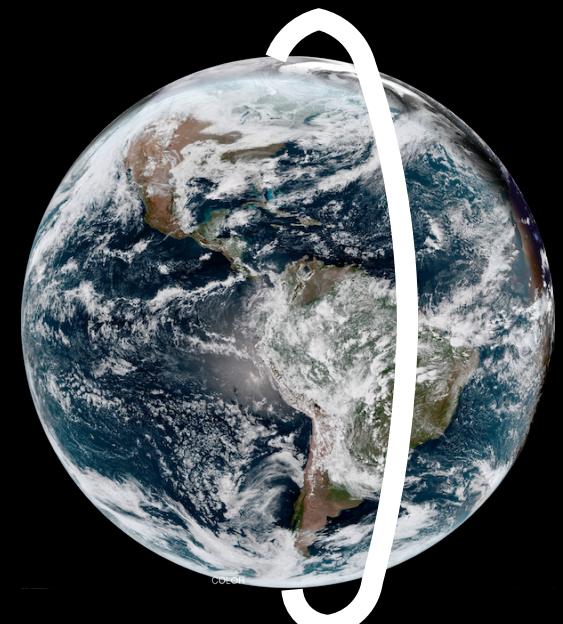
Example Subsystem Config. Files

Processor peripheral
running object
detection workload:

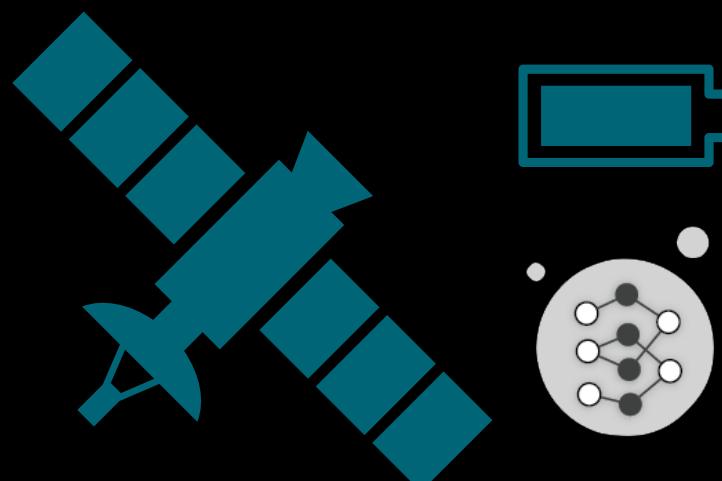
```
1 constant:task-duration-s:0.044860
2 variable:work-time-s:0.0
3 variable:power-w:0.0
4 variable:node-voltage:0.0
5 variable:claimed-task-count:0
6 variable:transmitting:0
7 enter-state:OFF:power-w=0.0
8 enter-state:SLEEP:power-w=0.5
9 enter-state:IDLE:power-w=0.5
10 enter-state:PAUSE:power-w=0.5
11 enter-state:WORK:power-w=11.3272
12 exit-state:WORK:work-time-s=0.0
13 transition:OFF->IDLE:node-voltage>=6.75,claimed-task-count==0
14 transition:OFF->PAUSE:node-voltage>=6.75,claimed-task-count!=0,transmitting!=0
15 transition:OFF->WORK:node-voltage>=6.75,claimed-task-count!=0,transmitting==0
16 transition:SLEEP->OFF:node-voltage<5.5
17 transition:SLEEP->IDLE:node-voltage>=6.75,claimed-task-count==0
18 transition:SLEEP->PAUSE:node-voltage>=6.75,claimed-task-count!=0,transmitting!=0
19 transition:SLEEP->WORK:node-voltage>=6.75,claimed-task-count!=0,transmitting==0
20 transition:IDLE->OFF:node-voltage<5.5
21 transition:IDLE->SLEEP:node-voltage<5.75,node-voltage>=5.5
22 transition:IDLE->PAUSE:node-voltage>=5.75,claimed-task-count!=0,transmitting!=0
23 transition:IDLE->WORK:node-voltage>=5.75,claimed-task-count!=0,transmitting==0
24 transition:PAUSE->OFF:node-voltage<5.5
25 transition:PAUSE->SLEEP:node-voltage<5.75,node-voltage>=5.5
26 transition:PAUSE->IDLE:node-voltage>=5.75,claimed-task-count==0
27 transition:PAUSE->WORK:node-voltage>=5.75,claimed-task-count!=0,transmitting==0
28 transition:WORK->OFF:node-voltage<5.5
29 transition:WORK->SLEEP:node-voltage<5.75,node-voltage>=5.5
30 transition:WORK->IDLE:node-voltage>=5.75,claimed-task-count==0
31 transition:WORK->PAUSE:node-voltage>=5.75,claimed-task-count!=0,transmitting!=0
32 initial-state:OFF
33
```

The cote Workflow

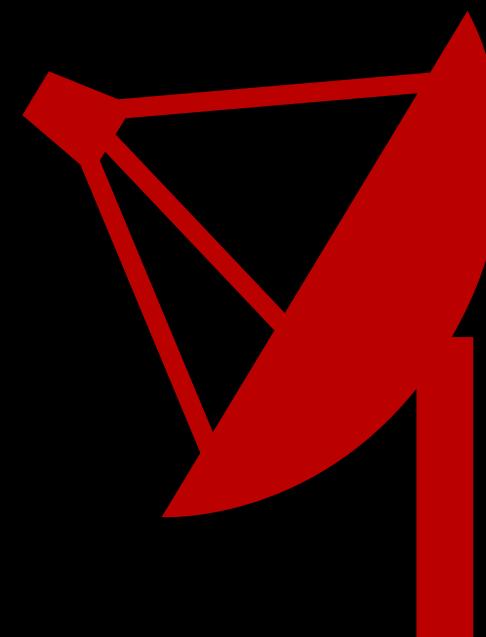
Simulator Inputs



Orbital Mechanics
Configuration



Satellite Configuration:
• Energy System
• Onboard Processing
• Other Peripherals



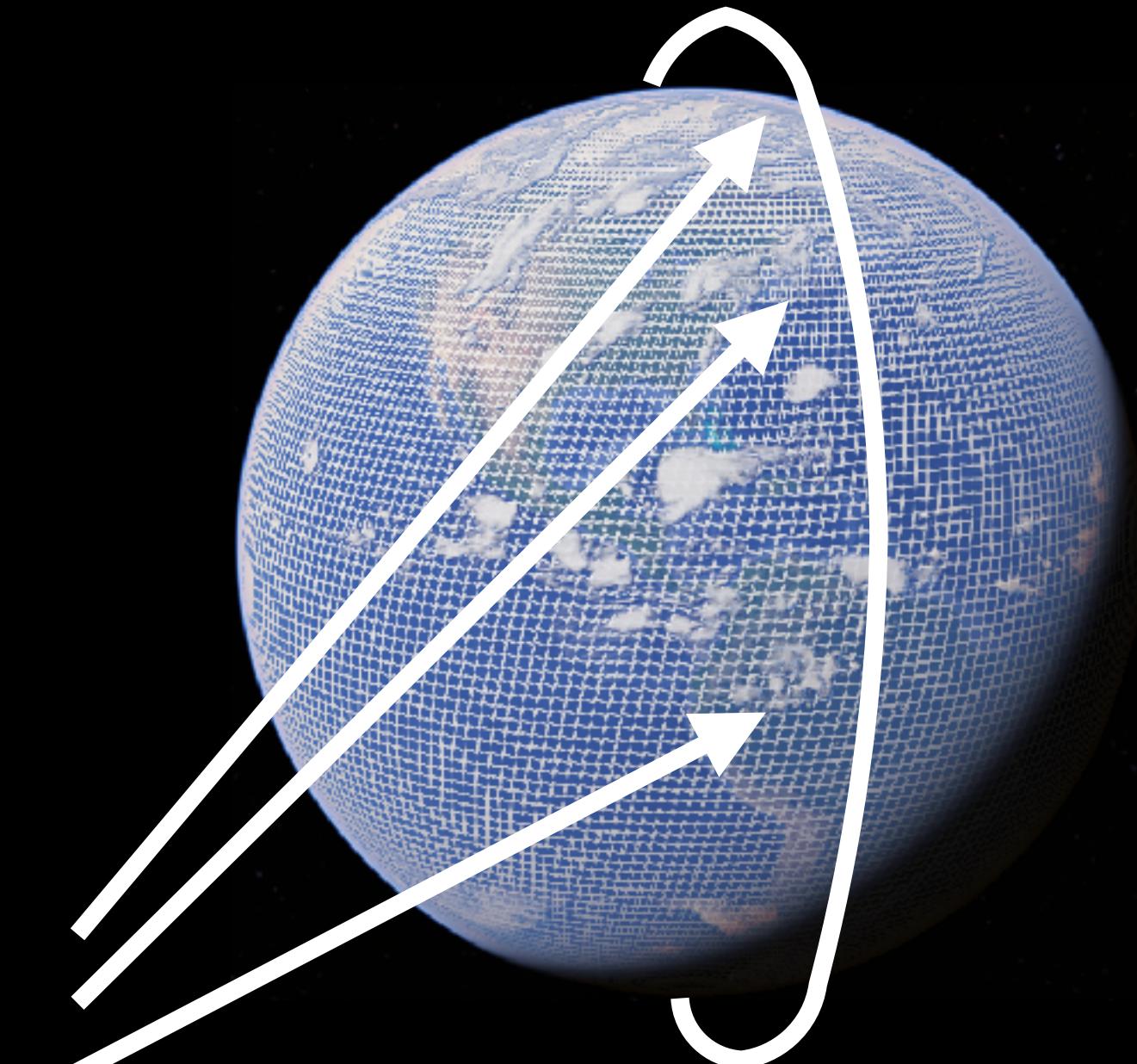
Communications
Configuration

Simulator Outputs

Scenario
Simulation



Event Traces



Observation Dates and Times
Satellite Activity Traces
Communication Logs

Example Output Log File

Log Date and Time Satellite ECI Position

	time, satellite-x, satellite-y, satellite-z
1	2019-07-12T19:56:47.771808000, 1071.933228, -1461.702515, 6635.830566
2	2019-07-12T19:57:47.771808000, 1120.461426, -1900.001221, 6515.723633
3	2019-07-12T19:58:47.771808000, 1164.053223, -2329.917969, 6366.841797
4	2019-07-12T19:59:47.771808000, 1202.513428, -2749.551514, 6189.834961
5	2019-07-12T20:00:47.771808000, 1235.674194, -3157.059570, 5985.477539
6	2019-07-12T20:01:47.771808000, 1263.385498, -3550.613281, 5754.679688
7	2019-07-12T20:02:47.771808000, 1285.523804, -3928.494873, 5498.444336
8	2019-07-12T20:03:47.771808000, 1301.990356, -4289.009277, 5217.918457
9	2019-07-12T20:04:47.771808000, 1312.711426, -4630.561035, 4914.339355
10	2019-07-12T20:05:47.771808000, 1317.637939, -4951.662109, 4589.023926
11	2019-07-12T20:06:47.771808000, 1316.746460, -5250.870117, 4243.421387
12	2019-07-12T20:07:47.771808000, 1310.040039, -5526.854980, 3879.067139
13	2019-07-12T20:08:47.771808000, 1297.548828, -5778.373535, 3497.604004
14	

Overview

Background and Related Work

Components of Space Simulation

The cote Workflow

Example Programs in cote

Open Source Development

<https://github.com/CMUAbstract/cote>

The screenshot shows the GitHub repository page for `CMUAbstract/cote`. The page has a dark theme. At the top, there's a navigation bar with links for `Pull requests`, `Issues`, `Marketplace`, and `Explore`. On the right side of the header are icons for notifications, a plus sign, and a user profile. Below the header, the repository name `CMUAbstract/cote` is displayed, along with a `Public` badge, an `Unwatch` button (with 5 watchers), a `Fork` button (with 1 fork), and a `Star` button (with 8 stars). A dropdown menu is open next to the star button. Below the header, there's a navigation bar with links for `Code`, `Issues` (1), `Pull requests`, `Actions`, `Projects`, `Wiki`, `Security`, `Insights`, and `Settings`. The `Code` link is underlined, indicating it's the active tab. On the left side, there's a sidebar showing the `master` branch (1 branch, 1 tag), a `Go to file` button, an `Add file` button, and a `Code` button (which is green). To the right of the sidebar, there's an `About` section with a brief description: "Orbital edge computing simulation software". Below the about section, there are links for `Readme`, `View license`, `8 stars`, `5 watching`, and `1 fork`. The main content area lists several commits by Brad Denby:

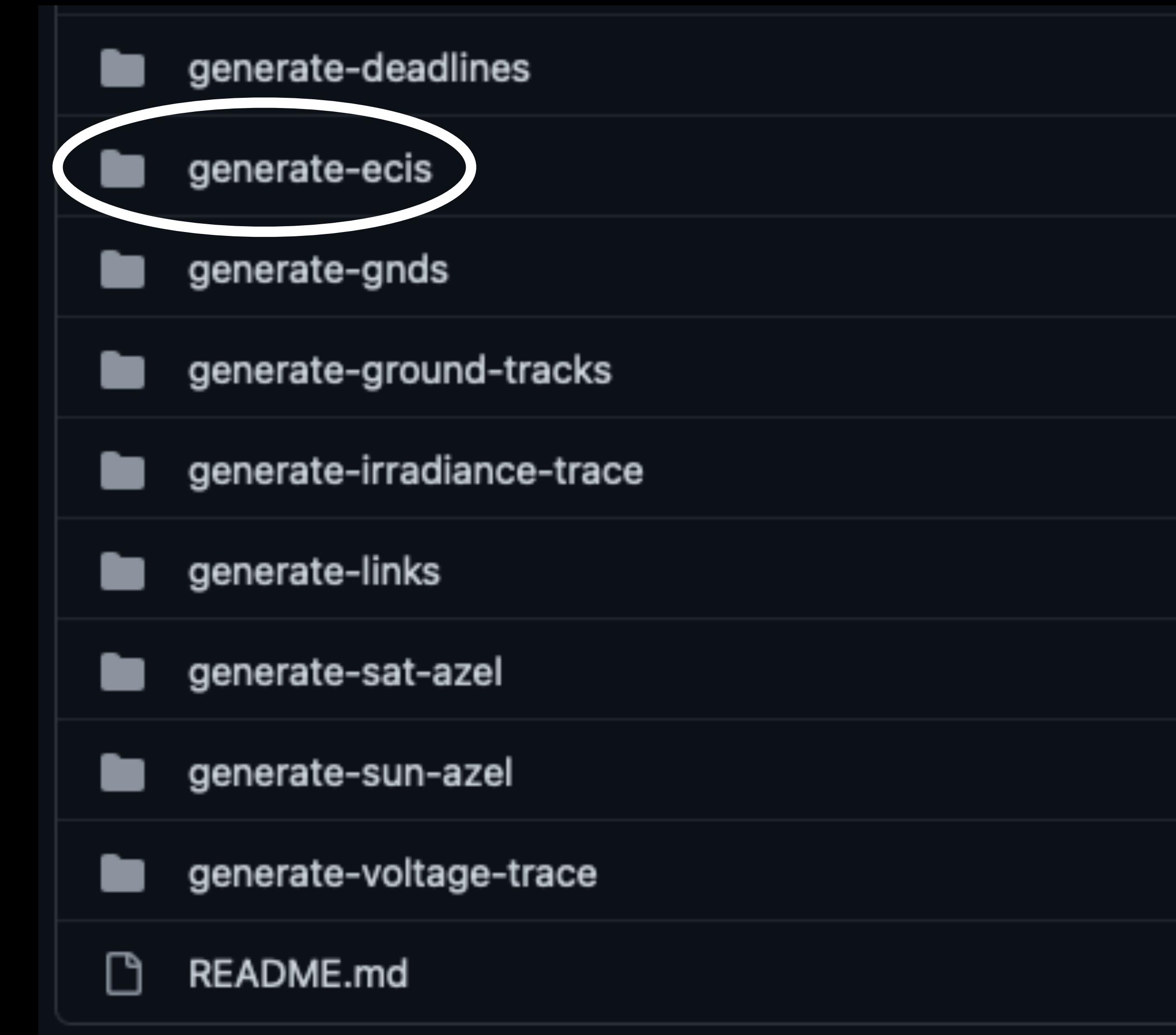
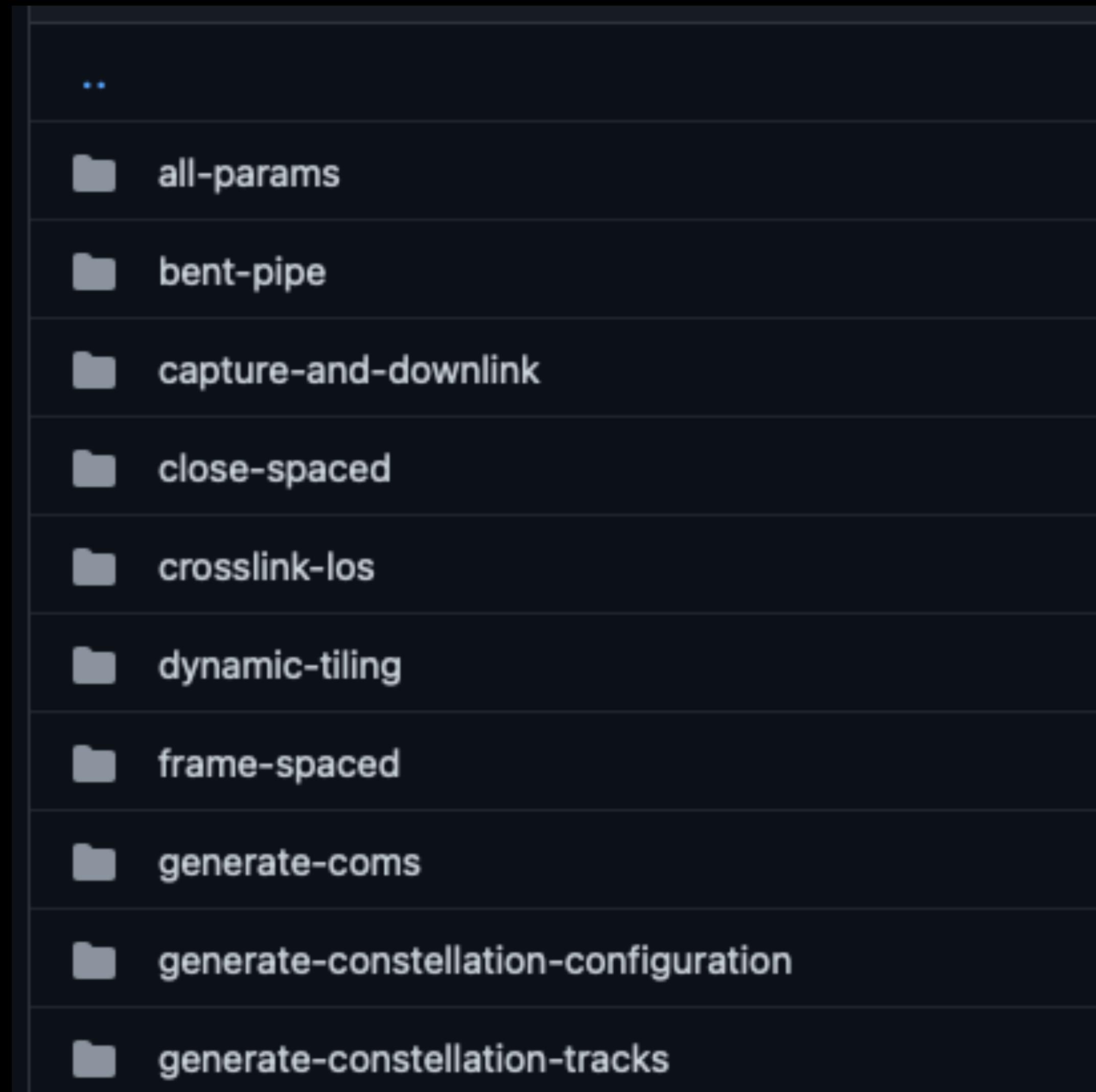
Commit	Message	Date
<code>analysis</code>	Adds analysis scripts for all params satellite positions	2 years ago
<code>dependencies/parse-json-0.1.0</code>	Adds all params parsing example	2 years ago
<code>examples</code>	Adds sample configuration files for capture and downlink example	3 months ago
<code>images</code>	cote v1.0.0 release	2 years ago
<code>references</code>	cote v1.0.0 release	2 years ago
<code>scripts</code>	Updates dependencies and prerequisites to avoid link rot	16 months ago
<code>software</code>	Adds crosslink line-of-sight utility function	17 months ago

Available Example Programs

..
all-params
bent-pipe
capture-and-downlink
close-spaced
crosslink-los
dynamic-tiling
frame-spaced
generate-coms
generate-constellation-configuration
generate-constellation-tracks

generate-deadlines
generate-ecis
generate-gnads
generate-ground-tracks
generate-irradiance-trace
generate-links
generate-sat-azel
generate-sun-azel
generate-voltage-trace
README.md

Available Example Programs



Example Program: “Generate ECIs”

Input Configuration Files

Output Log Files

Example Program: “Generate ECIs”

Input Configuration Files



Time Step: 1 minute

Output Log Files

Example Program: “Generate ECIs”

Input Configuration Files



Time Step: 1 minute



Step Count: 1440 (one day)

Output Log Files

Example Program: “Generate ECIs”

Input Configuration Files



Time Step: 1 minute



Step Count: 1440 (one day)



Simulation Start Date and Time

Output Log Files

Example Program: “Generate ECIs”

Input Configuration Files



Time Step: 1 minute



Step Count: 1440 (one day)



Simulation Start Date and Time



Satellite TLE

Output Log Files

Example Program: “Generate ECIs”

Input Configuration Files



Time Step: 1 minute



Step Count: 1440 (one day)



Simulation Start Date and Time



Satellite TLE



Output Log Files



**Satellite ECI x-position
(Once per minute)**

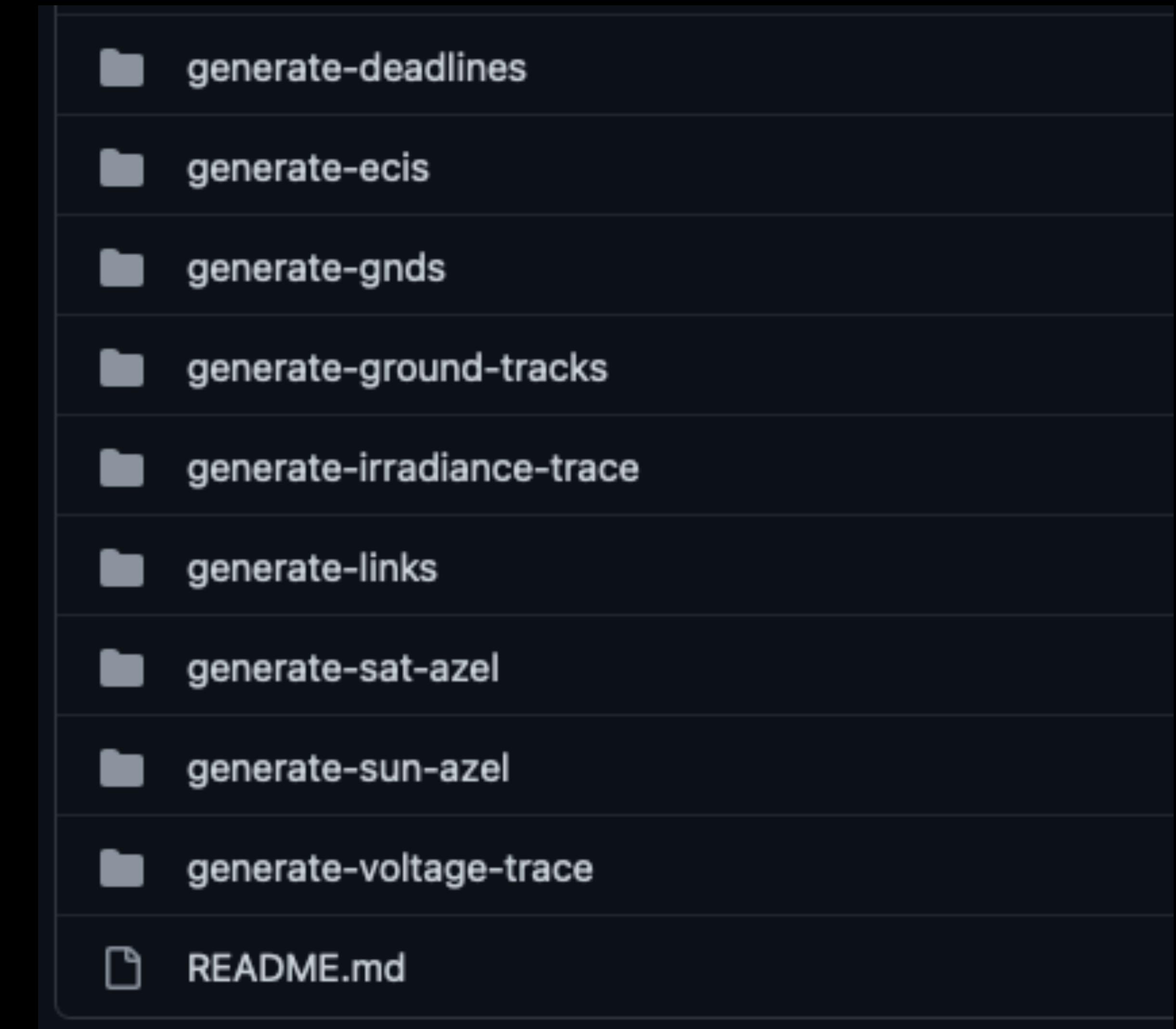
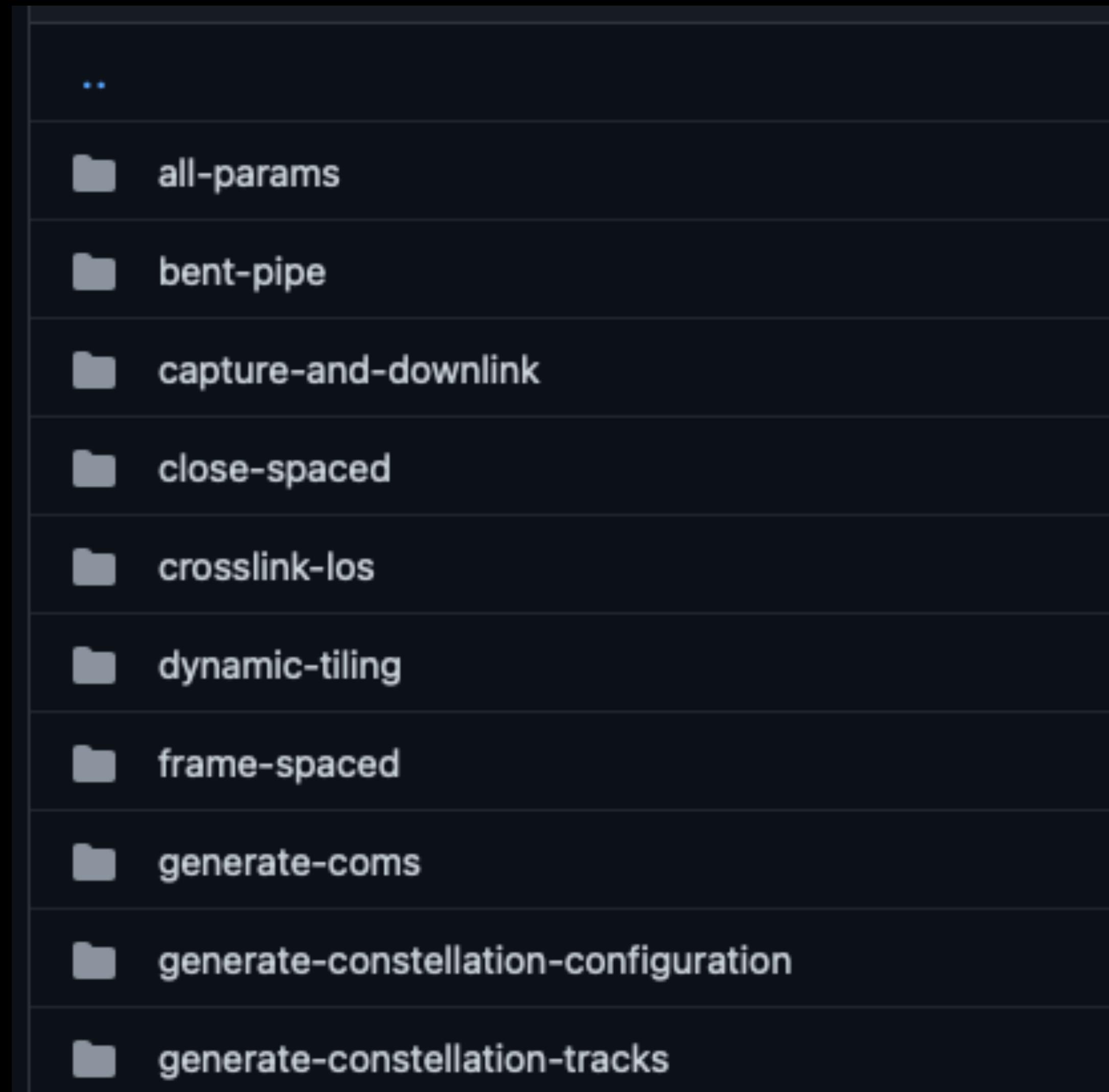


**Satellite ECI y-position
(Once per minute)**

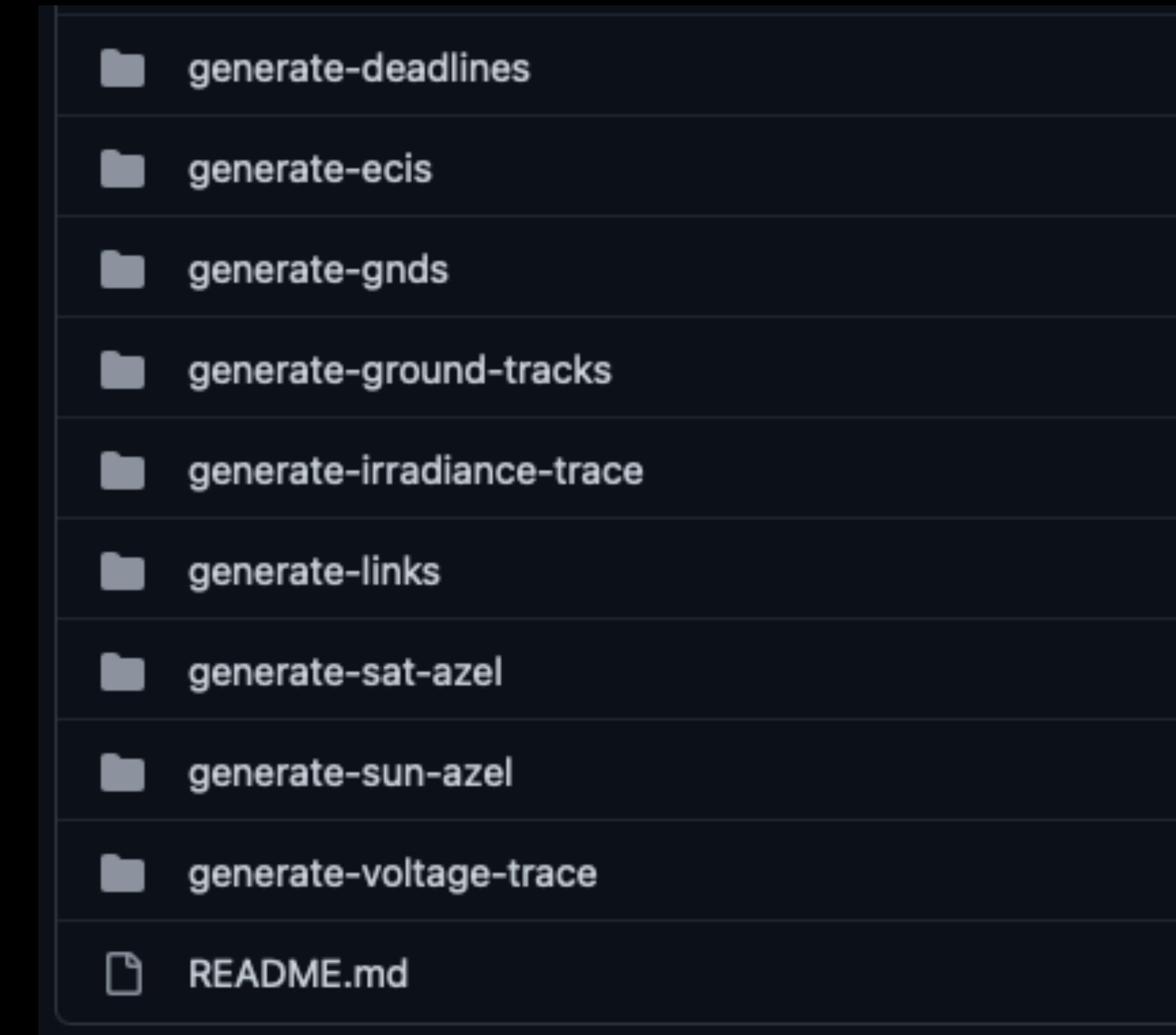
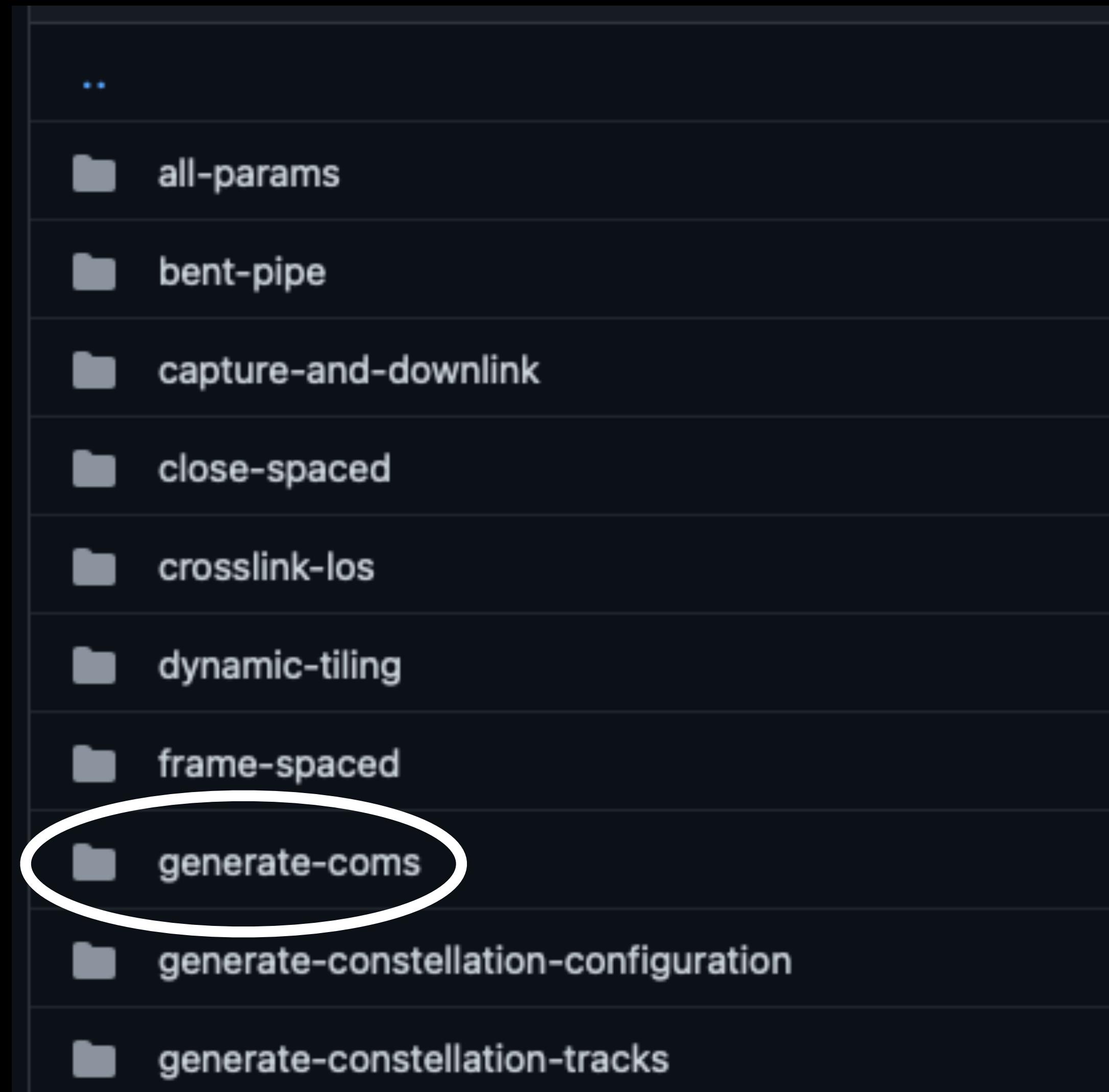


**Satellite ECI z-position
(Once per minute)**

Available Example Programs



Available Example Programs

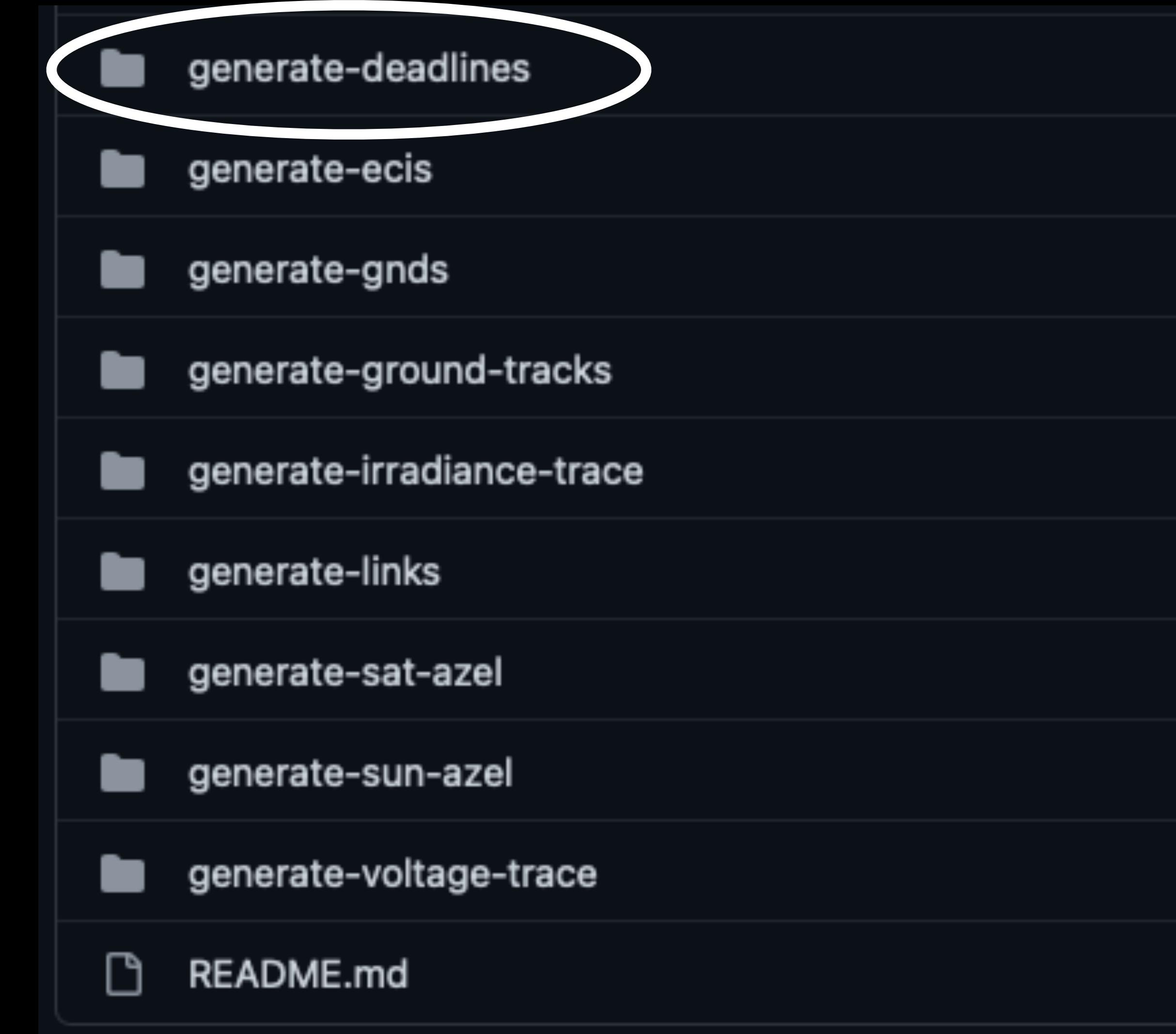
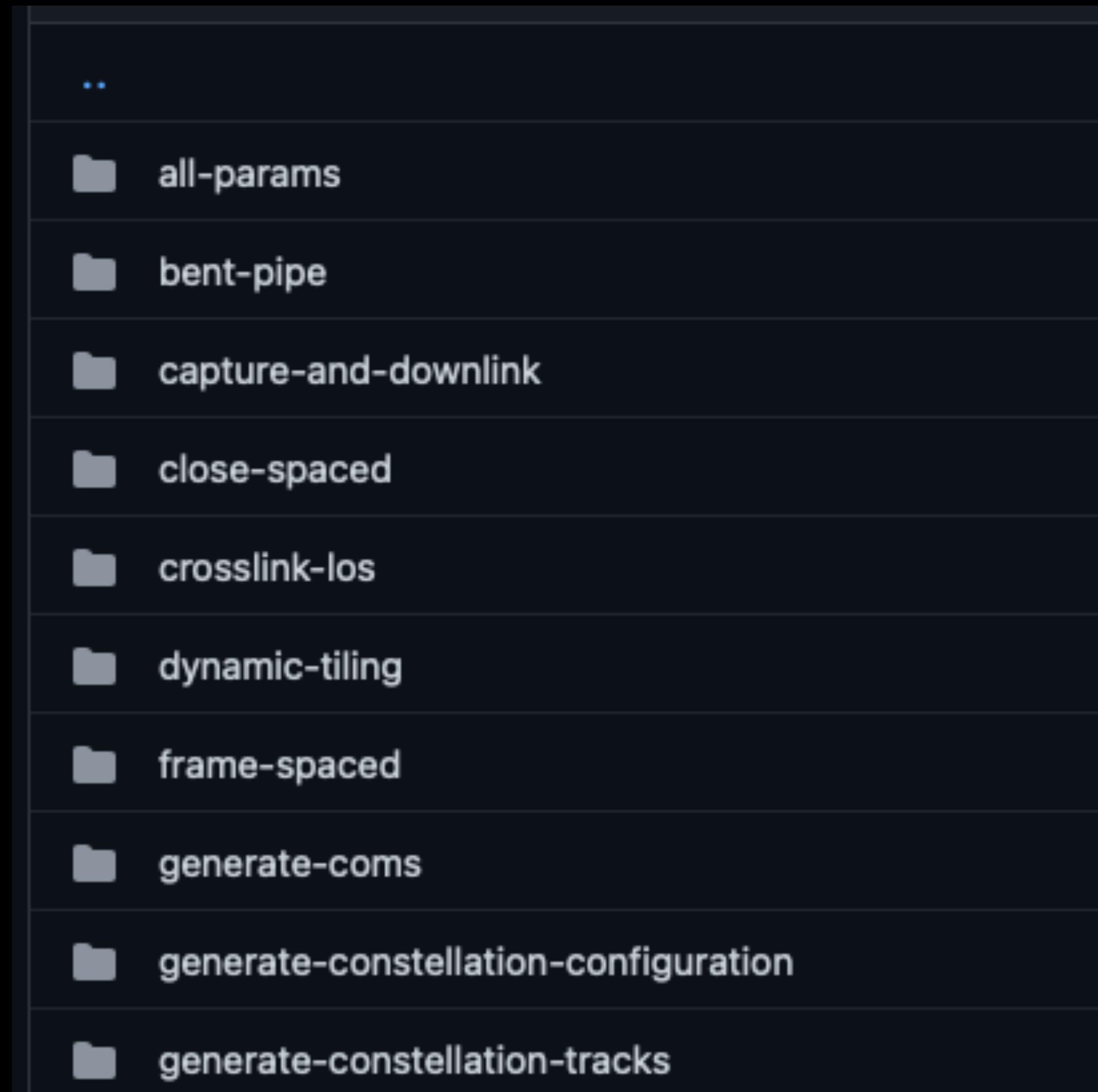


Available Example Programs

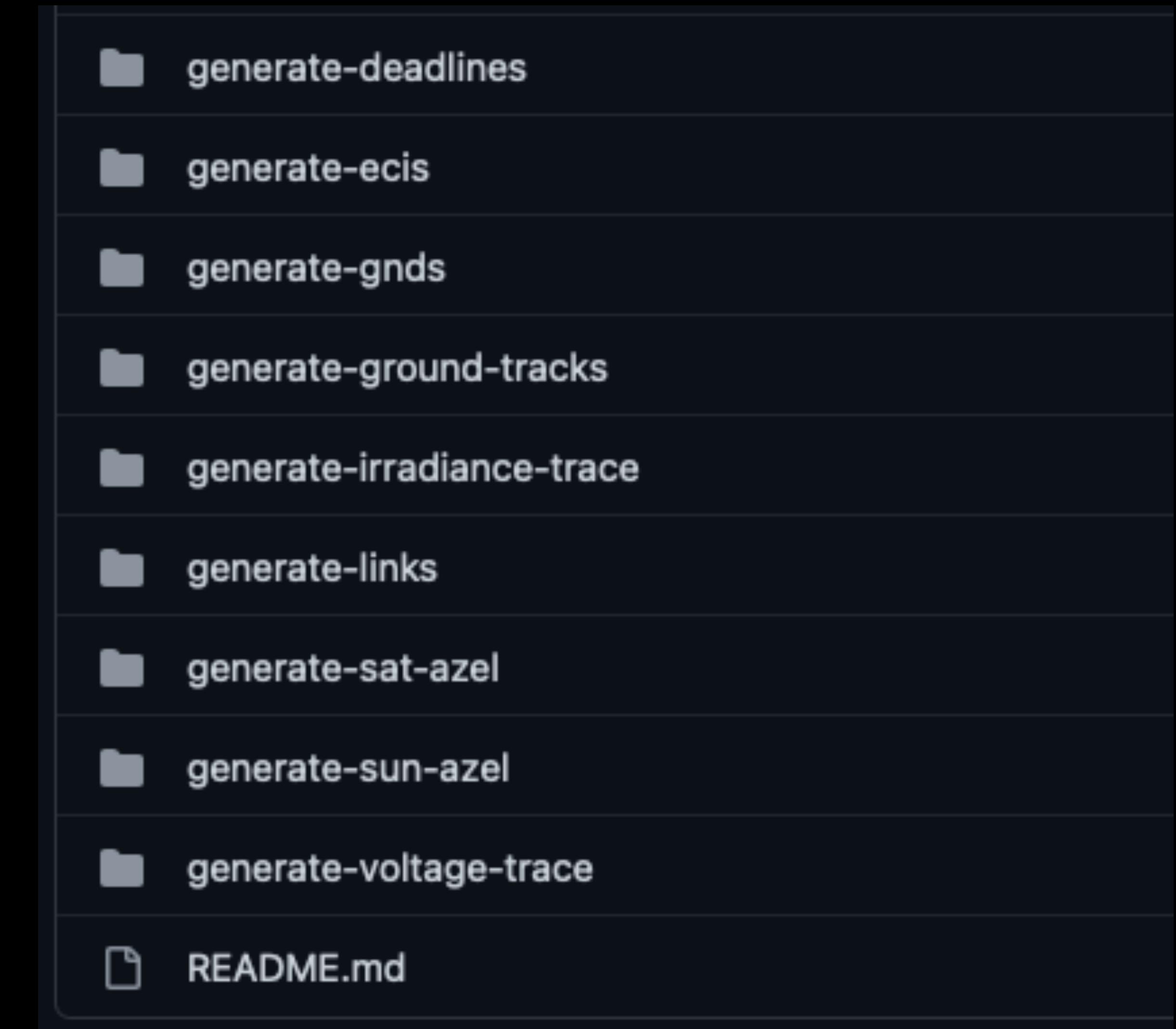
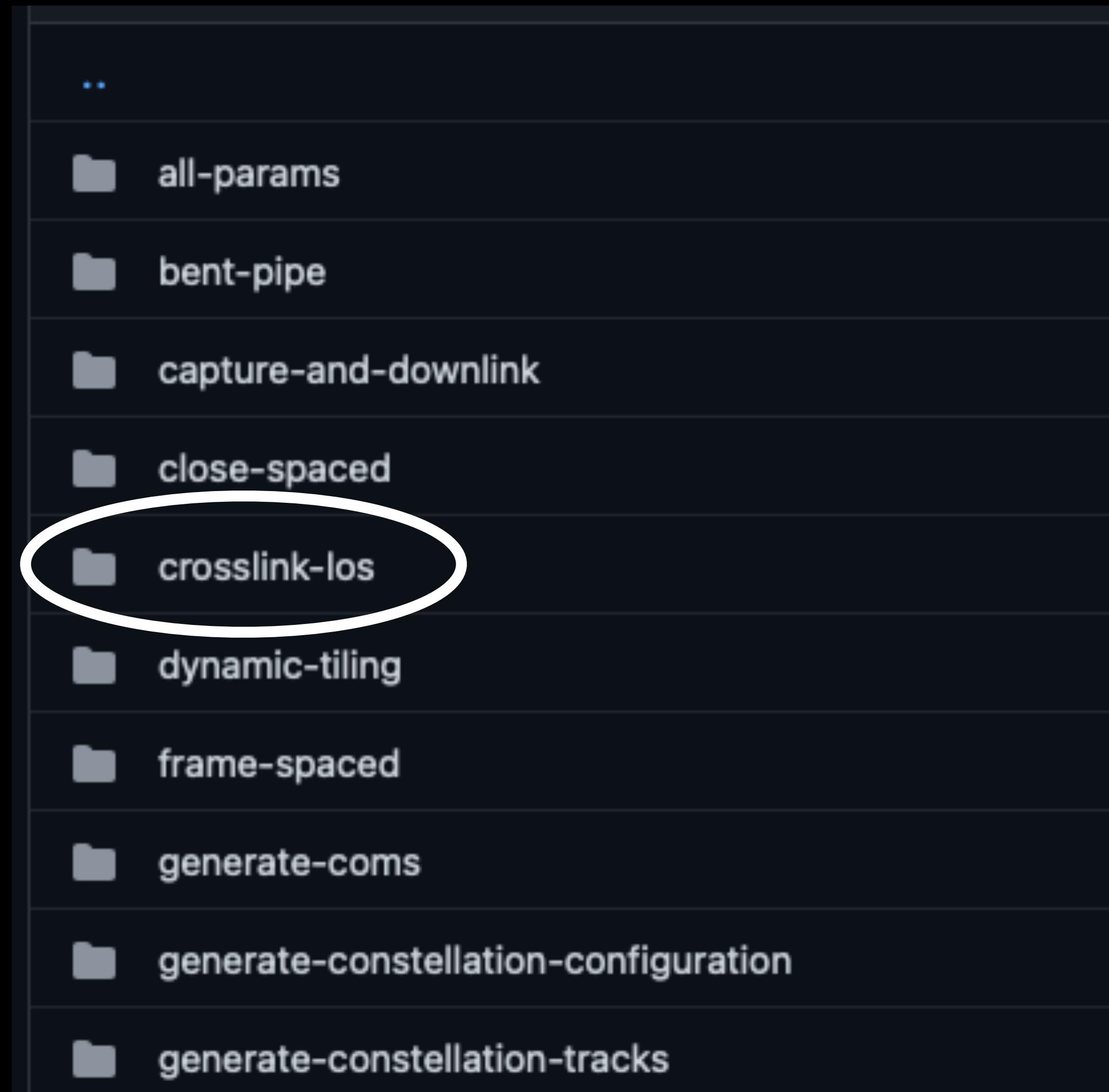
..
all-params
bent-pipe
capture-and-downlink
close-spaced
crosslink-los
dynamic-tiling
frame-spaced
generate-coms
generate-constellation-configuration
generate-constellation-tracks

generate-deadlines
generate-ecis
generate-gnads
generate-ground-tracks
generate-irradiance-trace
generate-links
generate-sat-azel
generate-sun-azel
generate-voltage-trace
README.md

Available Example Programs



Available Example Programs



Example Program: “Crosslink L.O.S.”

Input Configuration Files

Output Log Files

Example Program: “Crosslink L.O.S.”

Input Configuration Files



Time Step: 1 minute



Step Count: 49 (half orbit)



Simulation Start Date and Time



Many Satellite TLEs

Output Log Files

Example Program: “Crosslink L.O.S.”

Input Configuration Files



Time Step: 1 minute



Step Count: 49 (half orbit)

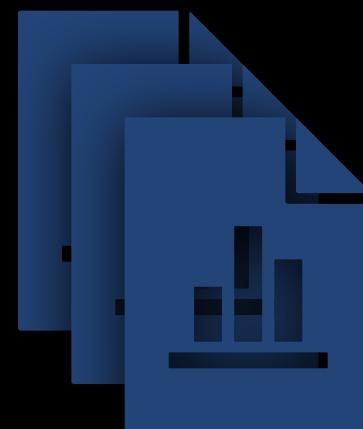


Simulation Start Date and Time

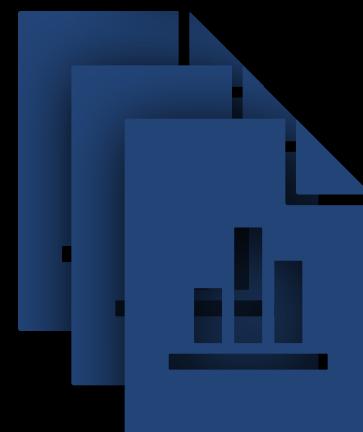


Many Satellite TLEs

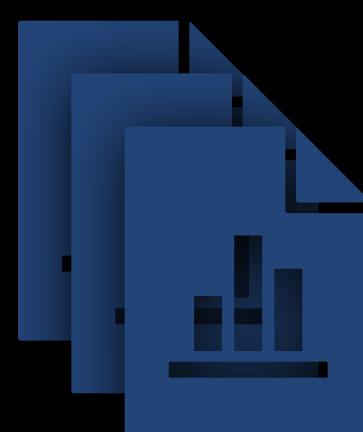
Output Log Files



Per-satellite link destination

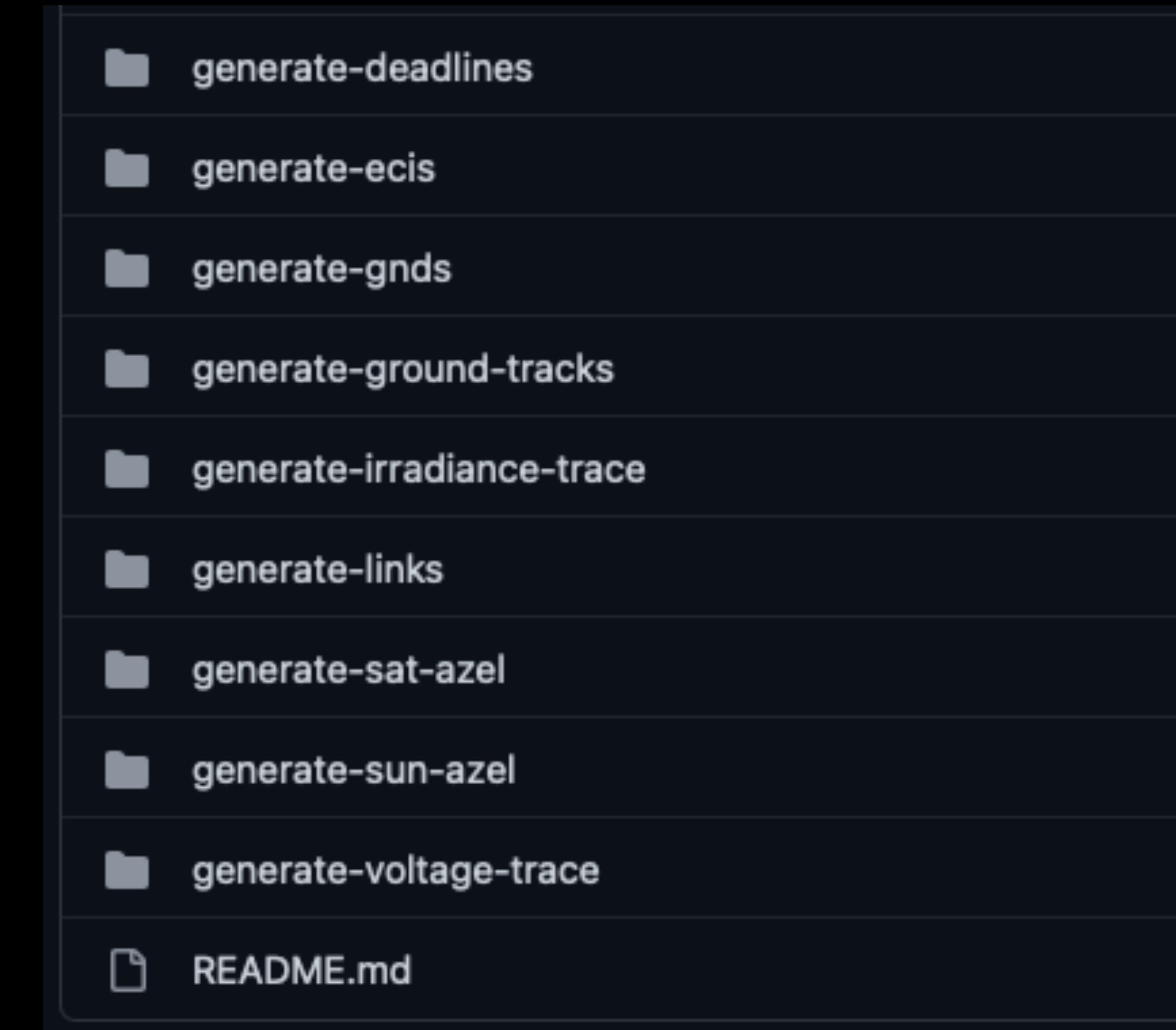
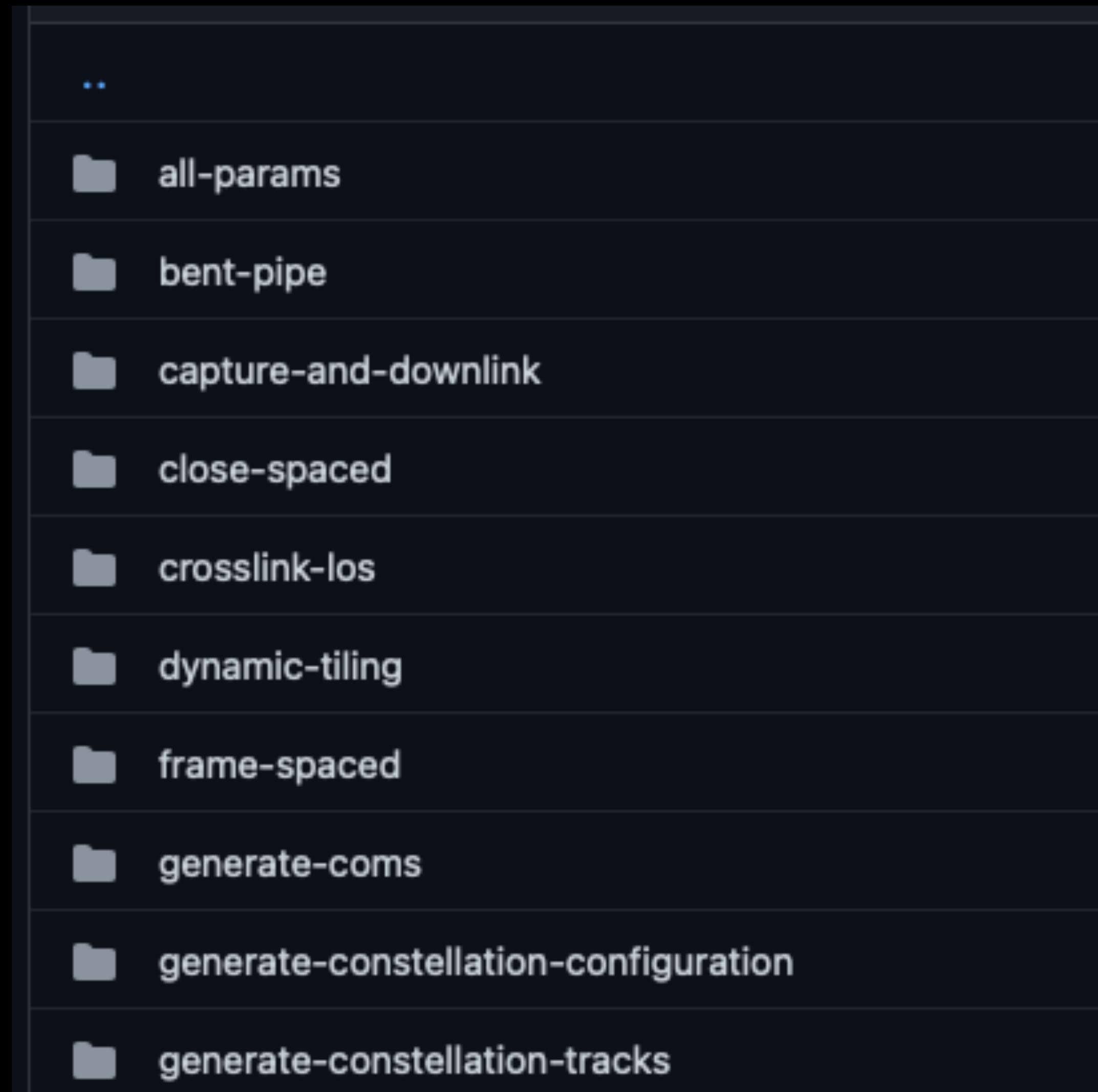


Per-satellite link distance

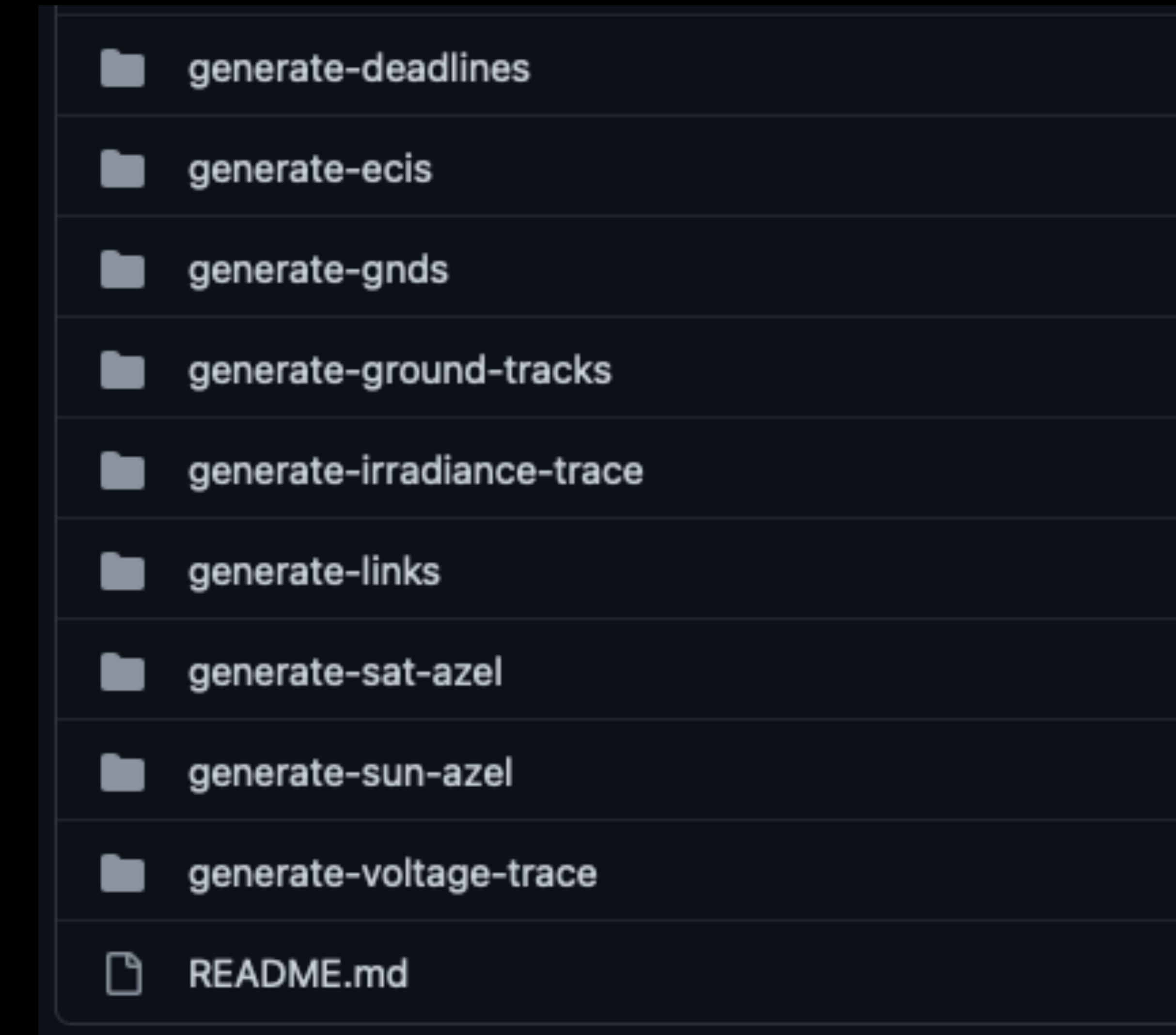
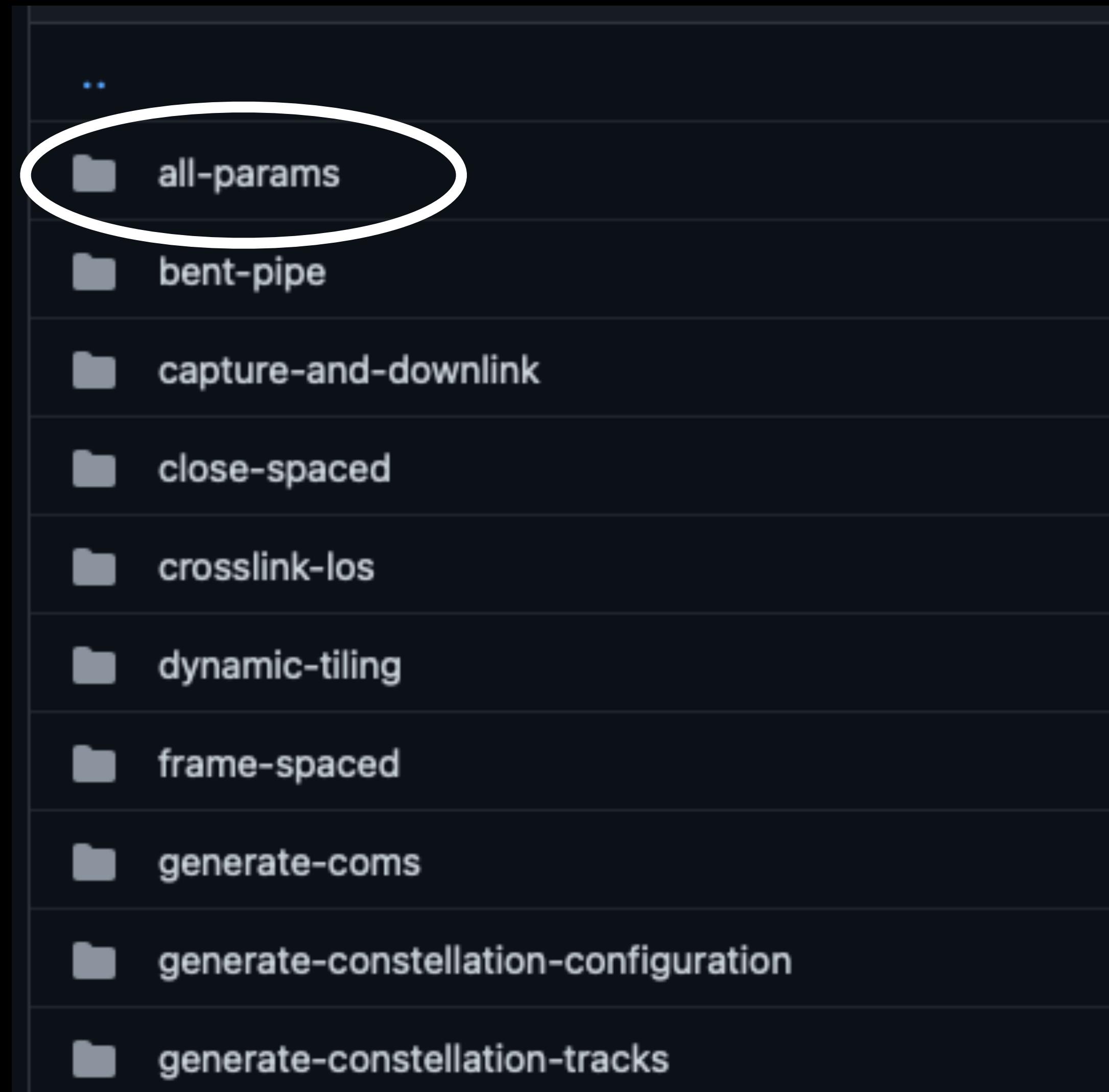


**Per-satellite ECI positions
(x, y, and z)**

Available Example Programs



Available Example Programs



Open Source Development

<https://github.com/CMUAbstract/cote>

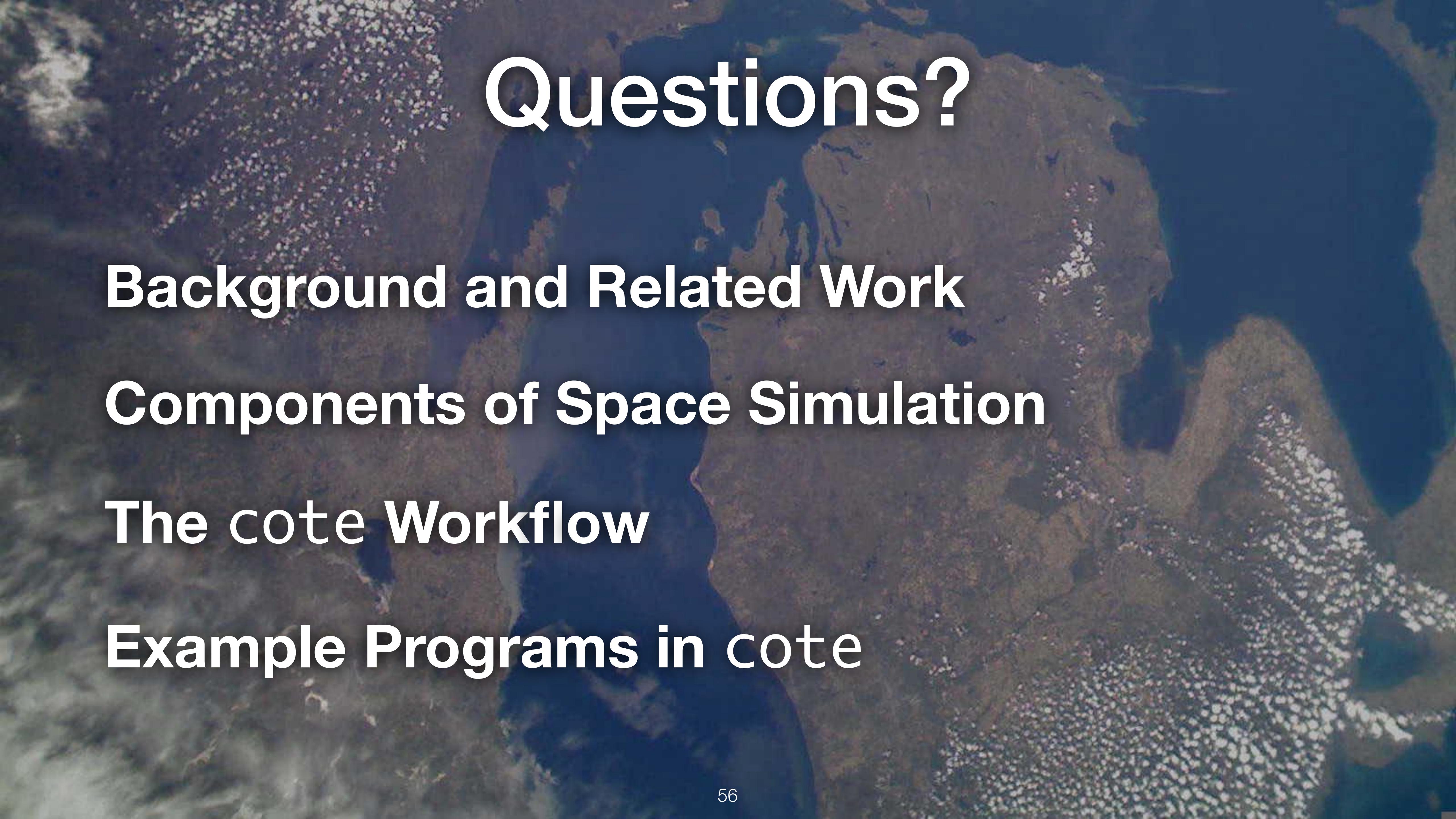
The screenshot shows the GitHub repository page for `cote`. The repository is owned by `CMUAbstract` and is public. It has 5 watchers, 1 fork, and 8 stars. The `Code` tab is selected. The repository has 1 branch and 1 tag. The latest commit was made by `Brad Denby` on July 19, 2019, with 65 commits. The commit message is "Adds sample configuration files for capture and downlink example". The commit hash is `3a3e4c9`. Below the commit, there is a list of files and their descriptions:

File	Description	Date
<code>analysis</code>	Adds analysis scripts for all params satellite positions	2 years ago
<code>dependencies/parse-json-0.1.0</code>	Adds all params parsing example	2 years ago
<code>examples</code>	Adds sample configuration files for capture and downlink example	3 months ago
<code>images</code>	cote v1.0.0 release	2 years ago
<code>references</code>	cote v1.0.0 release	2 years ago
<code>scripts</code>	Updates dependencies and prerequisites to avoid link rot	16 months ago
<code>software</code>	Adds crosslink line-of-sight utility function	17 months ago

On the right side, there is an **About** section with the following details:

- Orbital edge computing simulation software
- [Readme](#)
- [View license](#)
- 8 stars
- 5 watching
- 1 fork

At the bottom, there is a **Releases** section with 1 tag.

The background of the slide is a photograph of a coastal area from an aerial perspective. The land is covered in green vegetation, with a winding riverbed and a paved road visible. The ocean is a dark blue color.

Questions?

Background and Related Work

Components of Space Simulation

The cote Workflow

Example Programs in cote



cote

Software Simulation of the Space Environment

Brad Denby, bdenby@cmu.edu

Advisor: Brandon Lucia, blucia@andrew.cmu.edu

<https://github.com/CMUAbstract/cote>

Carnegie Mellon University