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

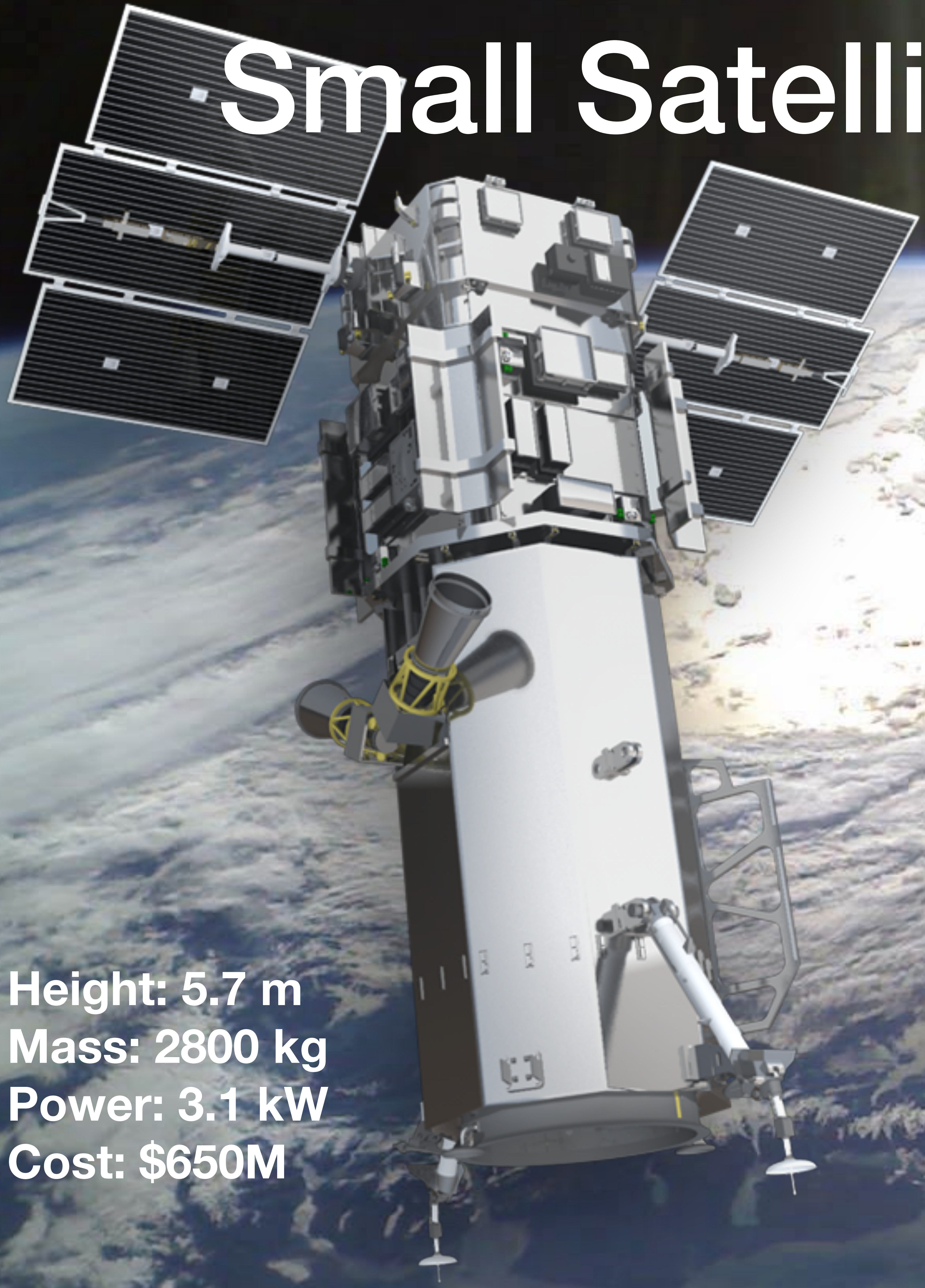
Background and Related Work

Components of Space Simulation

The `cote` Workflow

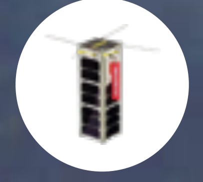
Example Programs in `cote`

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

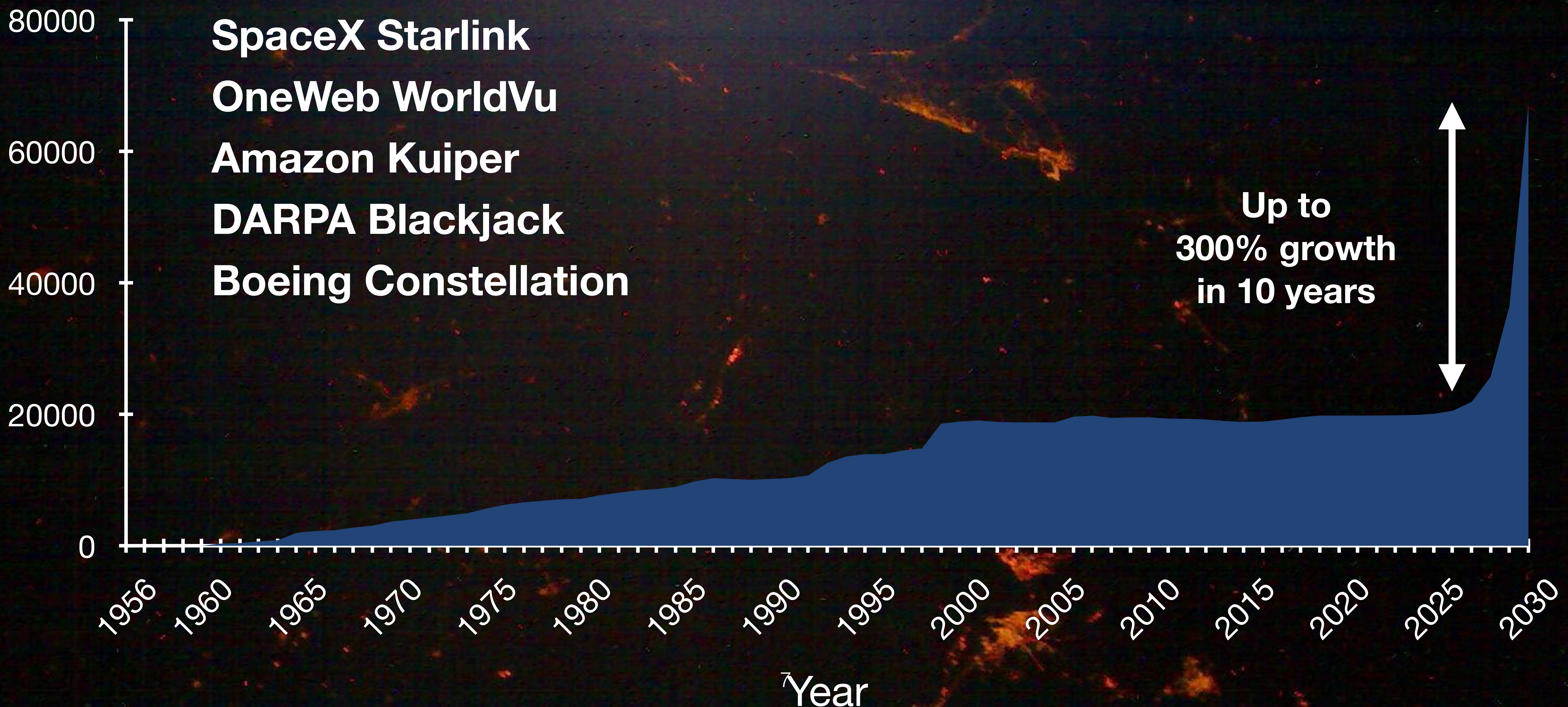


Pocketqubes and
Chip-scale Satellites

Little Innovation in CONOPS

CONOPS Must Change

Satellites Orbiting Earth



Downlinking Data Does Not Scale

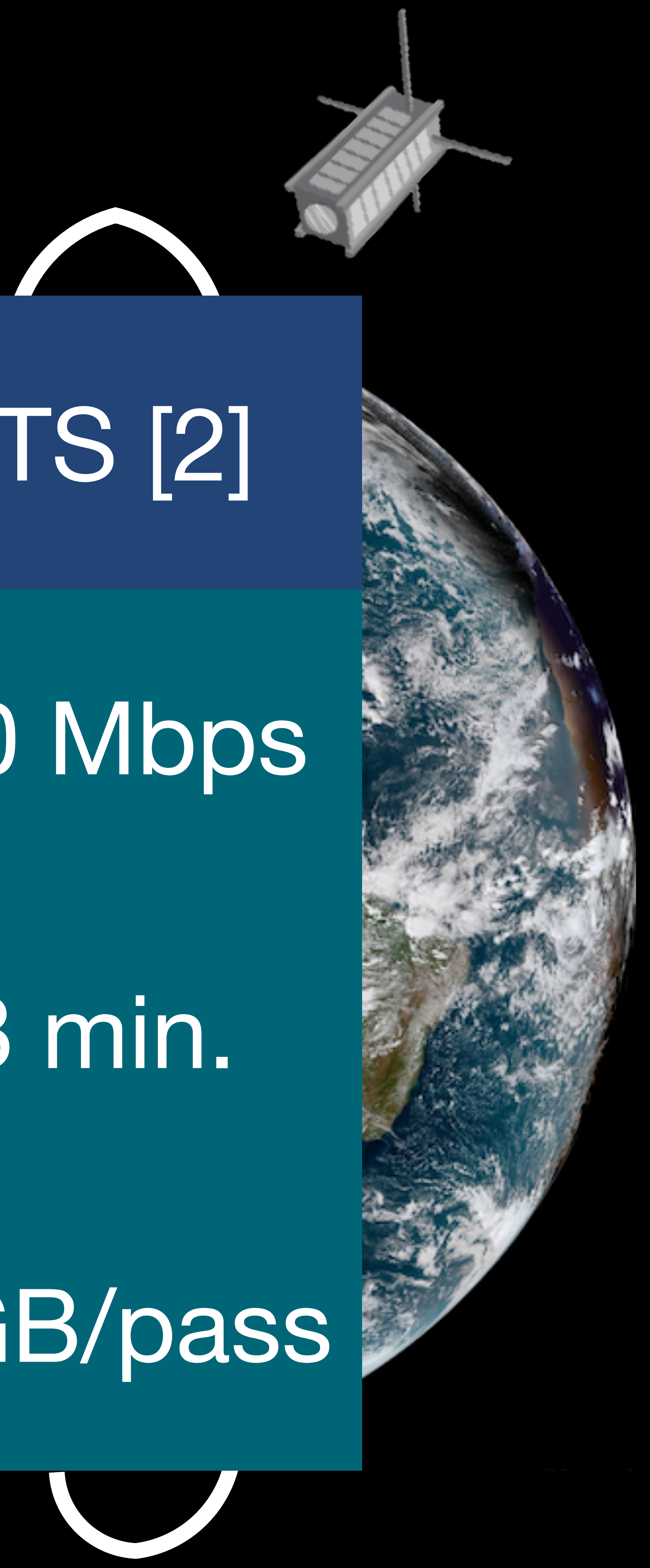
Downlinking Data Does Not Scale

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

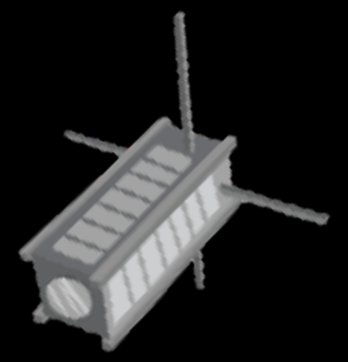


	Enterprise [1]	COTS [2]
Bitrate	200 Mbps	<120 Mbps
TX Duration	<10 min.	7-8 min.
Data Down	15 GB/pass	5.3 GB/pass

[1] Dove High Speed Downlink System

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

Downlinking Data Does Not Scale

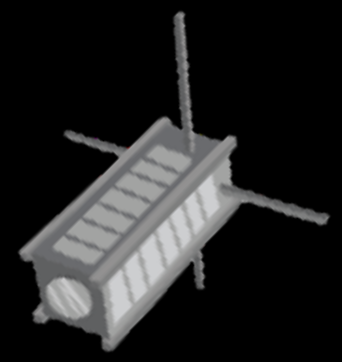


	Enterprise [1]	COTS [2]	Near Future
Bitrate	200 Mbps	<120 Mbps	1 Gbps?
TX Duration	<10 min.	7-8 min.	10 min.
Data Down	15 GB/pass	5.3 GB/pass	75 GB/pass

[1] Dove High Speed Downlink System

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

Downlinking Data Does Not Scale

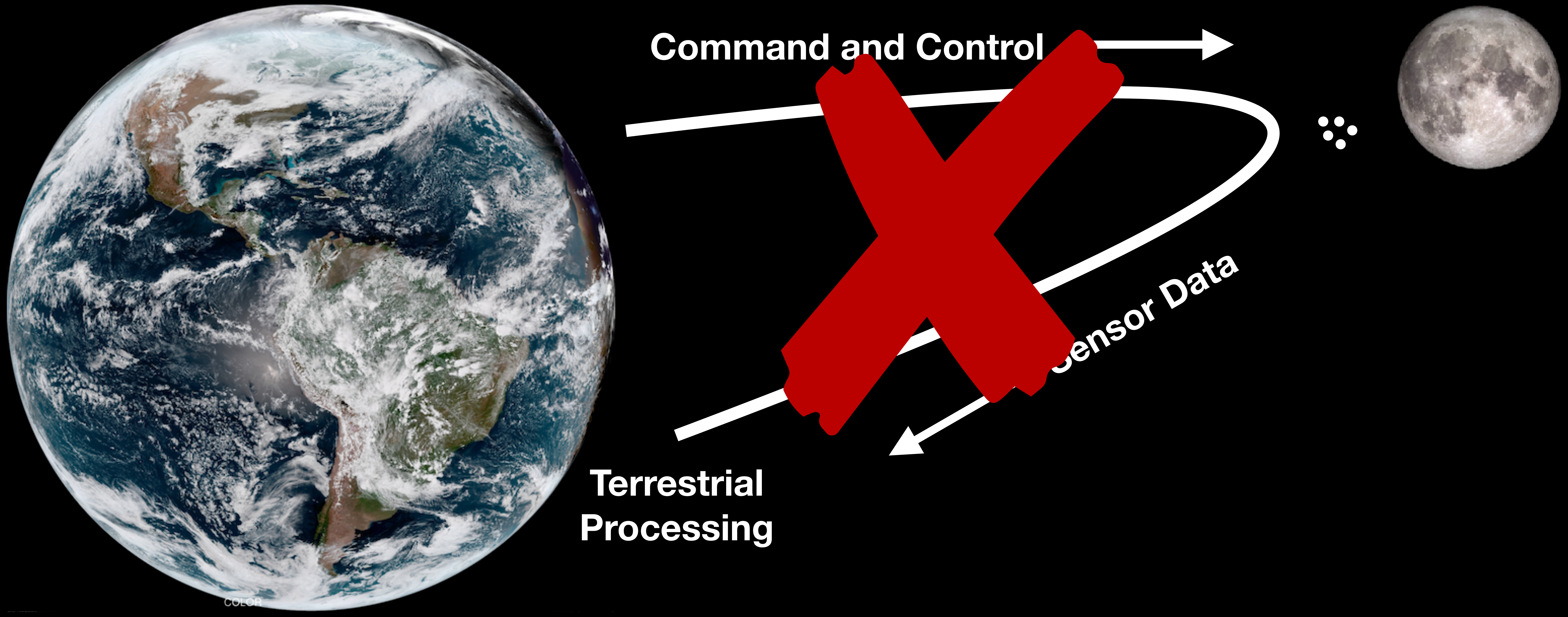


	Enterprise [1]	COTS [2]	Near Future	Mega Constellation
Bitrate	200 Mbps	<120 Mbps	1 Gbps?	No Change
TX Duration	<10 min.	7-8 min.	10 min.	↓(contention)
Data Down	15 GB/pass	5.3 GB/pass	75 GB/pass	Less per sat.

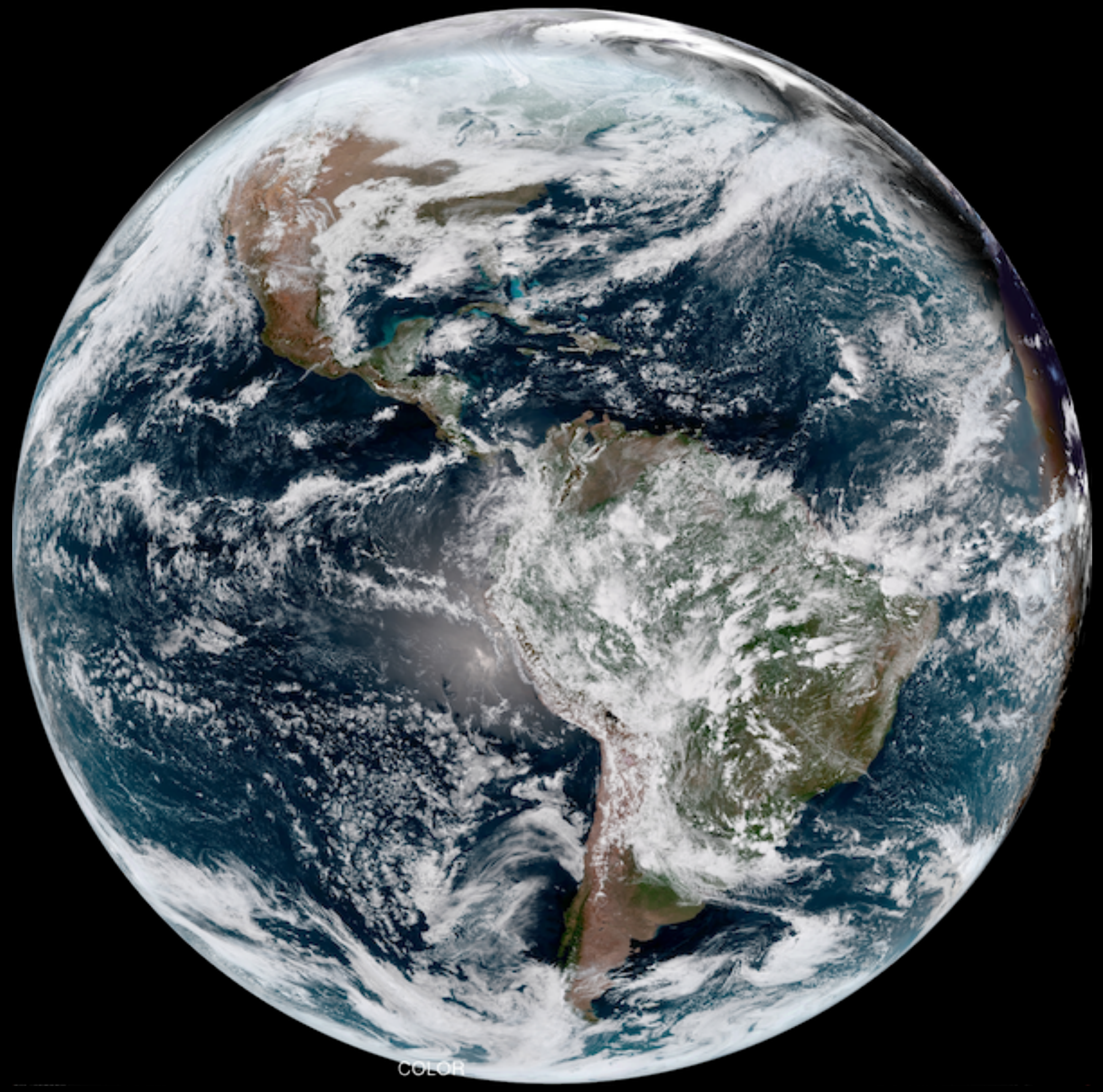
[1] Dove High Speed Downlink System

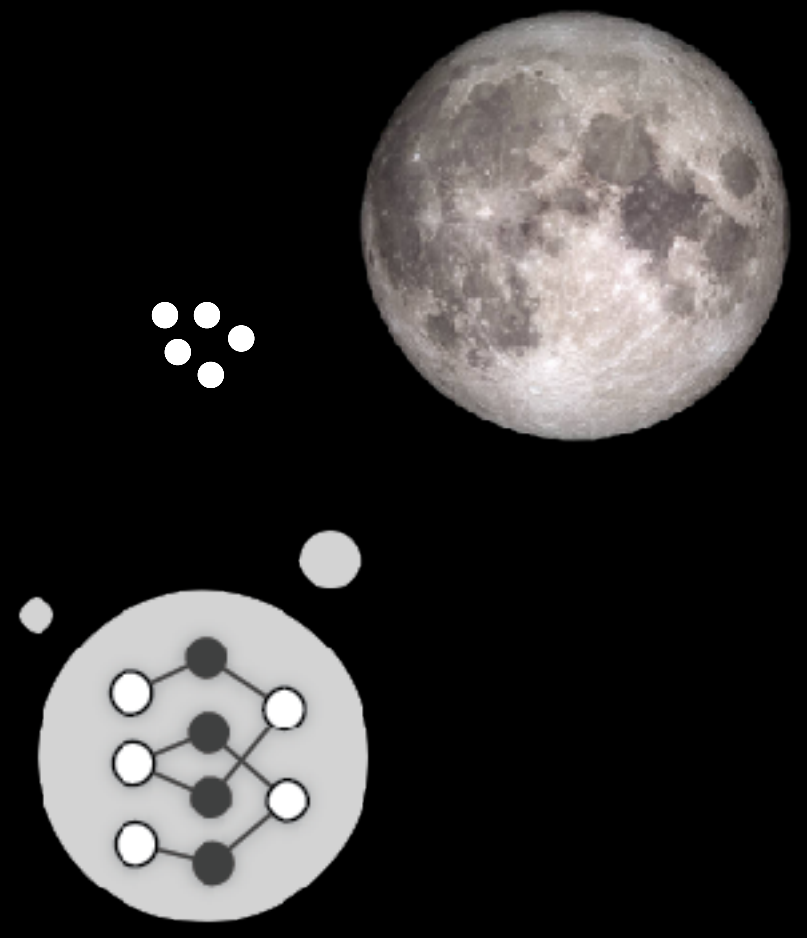
[2] `cote` Simulator: https://github.com/cmuabstract/cote

Reducing Reliance on Bent Pipes



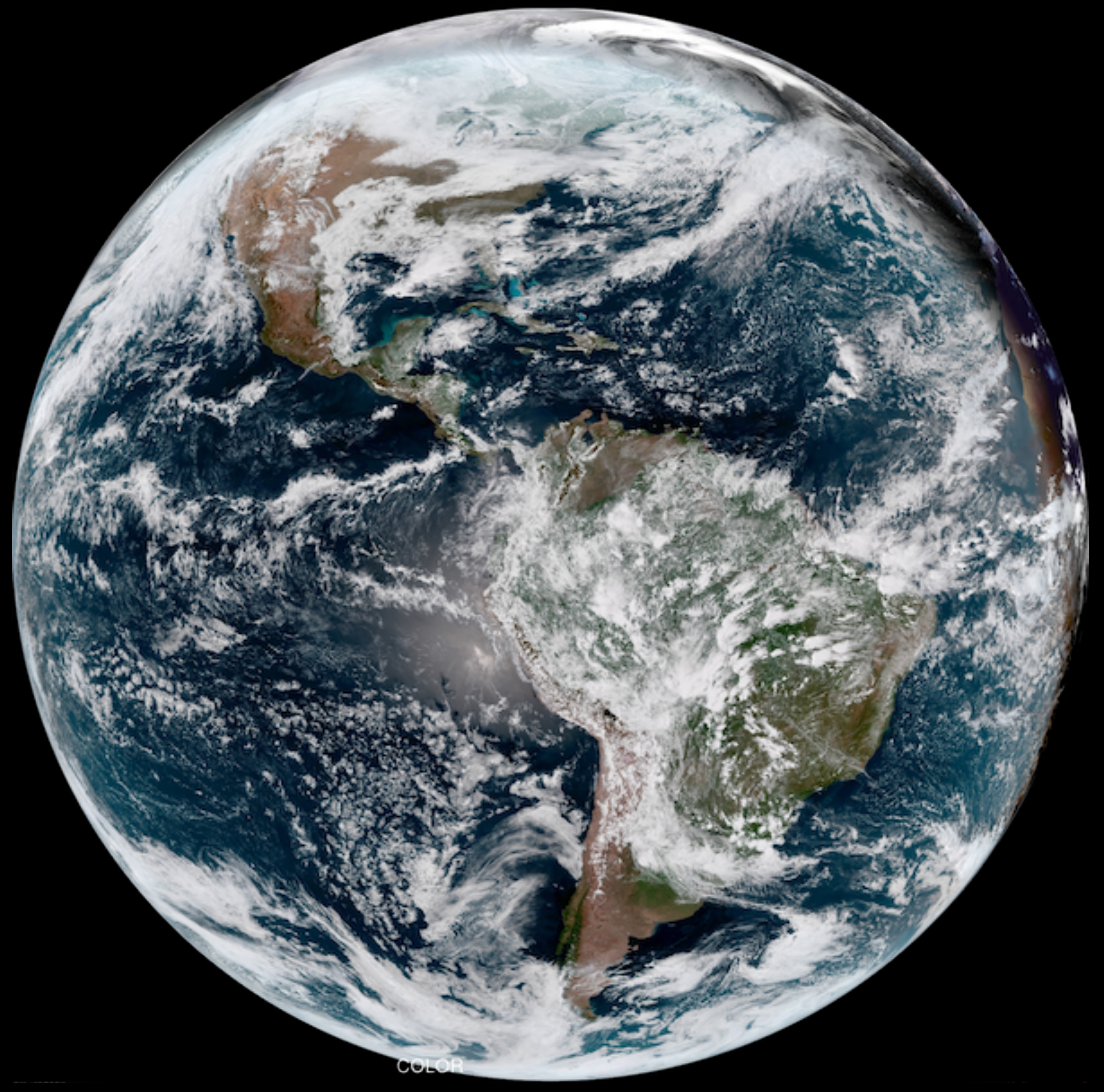
Edge Computing: A Scalable Alternative





Computing
at the Edge

Edge Computing: A Scalable Alternative

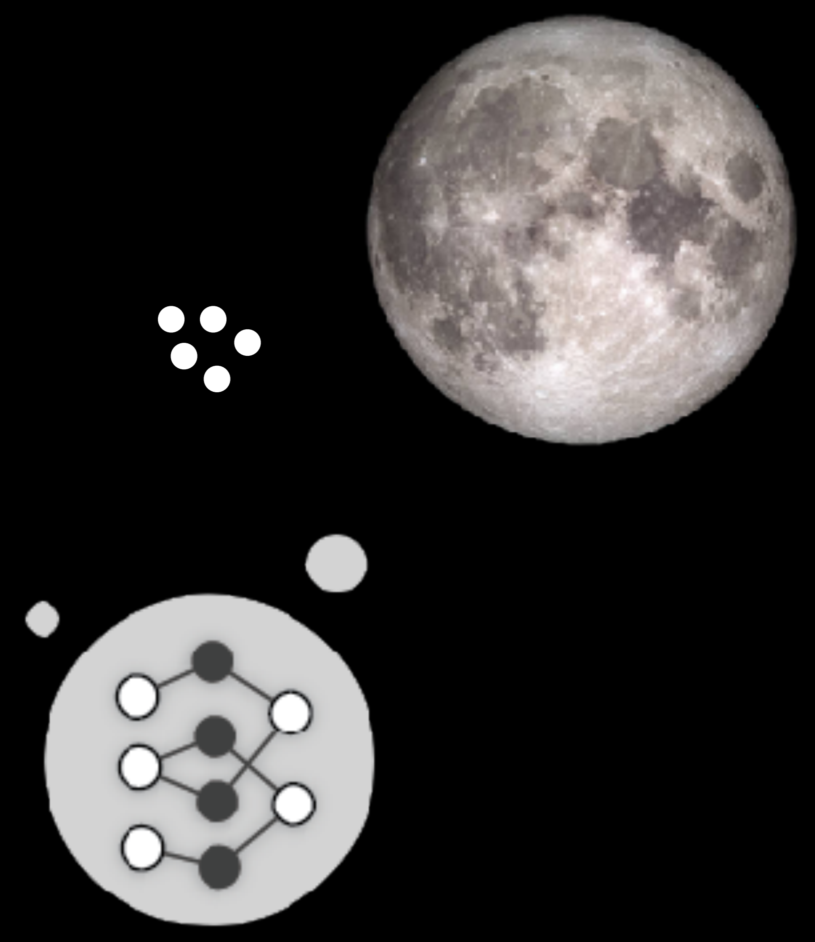


Existing Systems

Communication

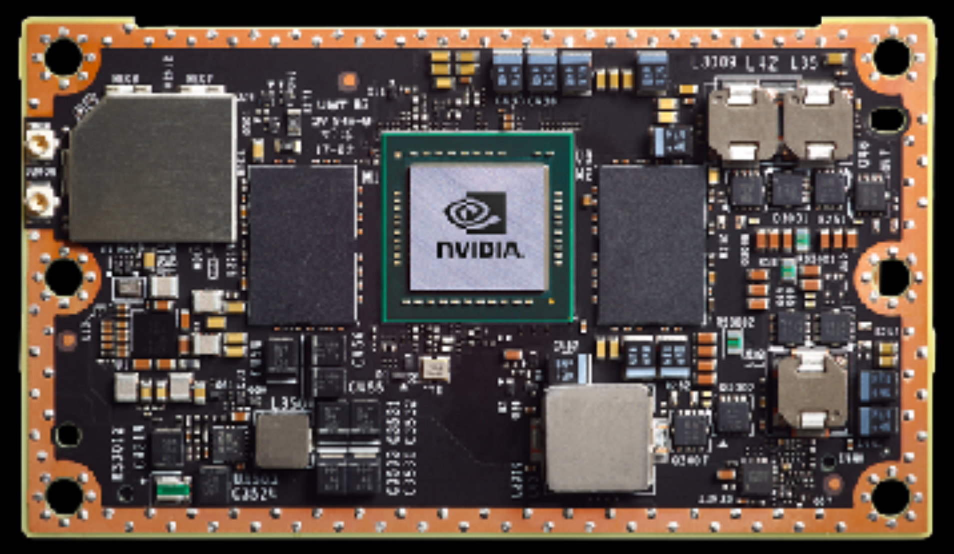
Guidance,
Navigation,
& Control

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

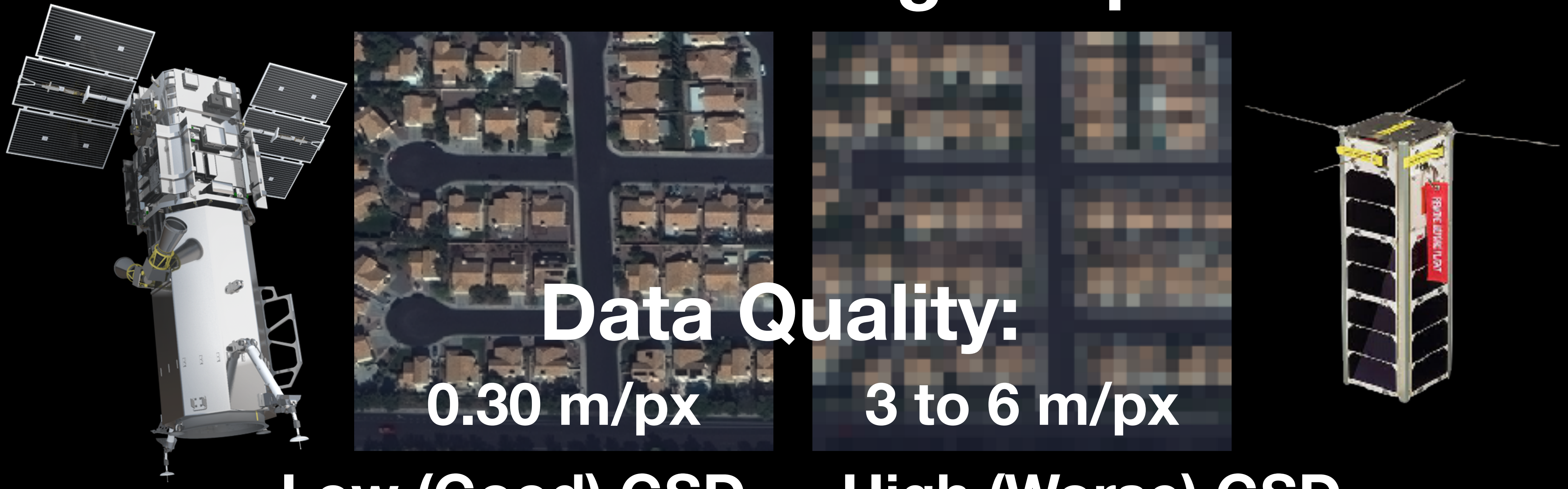
Low (Good) GSD **High (Worse) GSD**

Machine Learning Bridges the Gap

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

Computational Nanosatellites

Machine Learning in Space

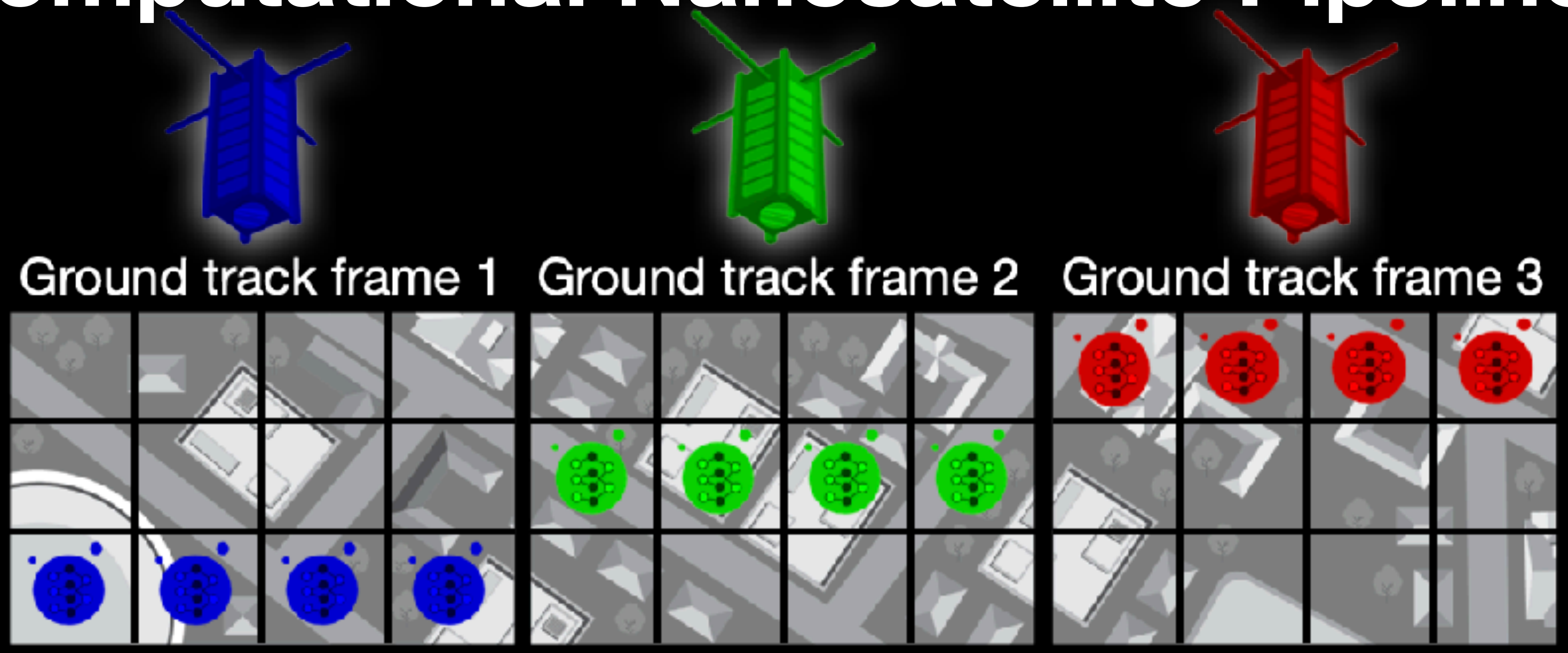


Low (Good) GSD High (Worse) GSD

Machine Learning Bridges the Gap

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

Computational Nanosatellite Pipelines

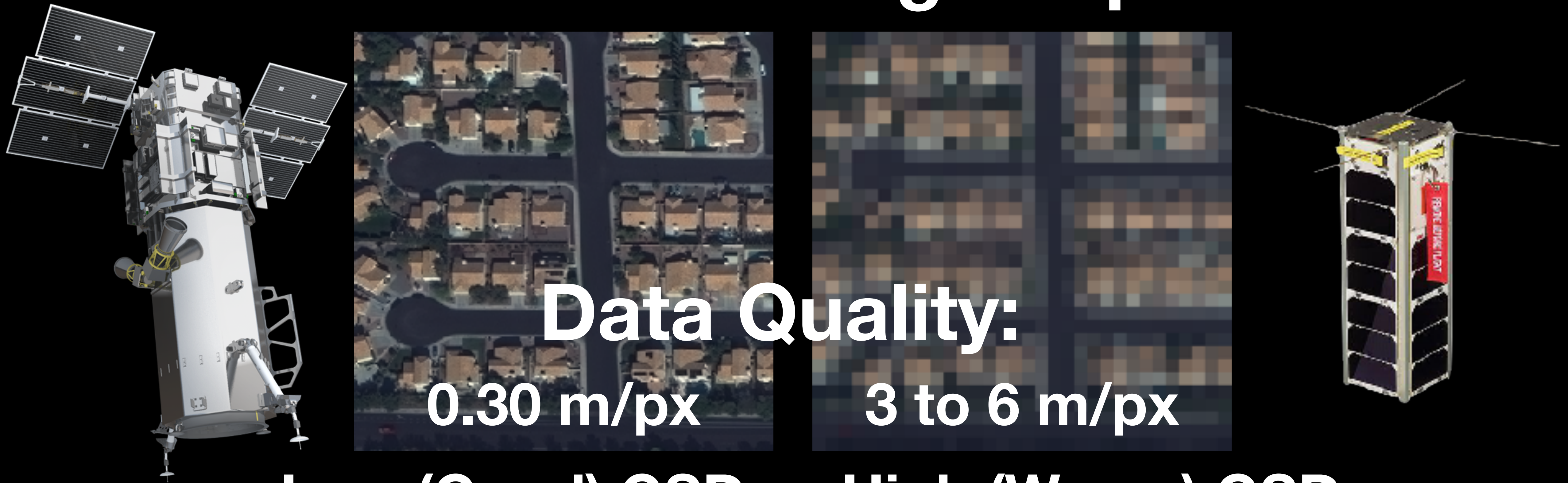


Distributing Computation across Devices

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

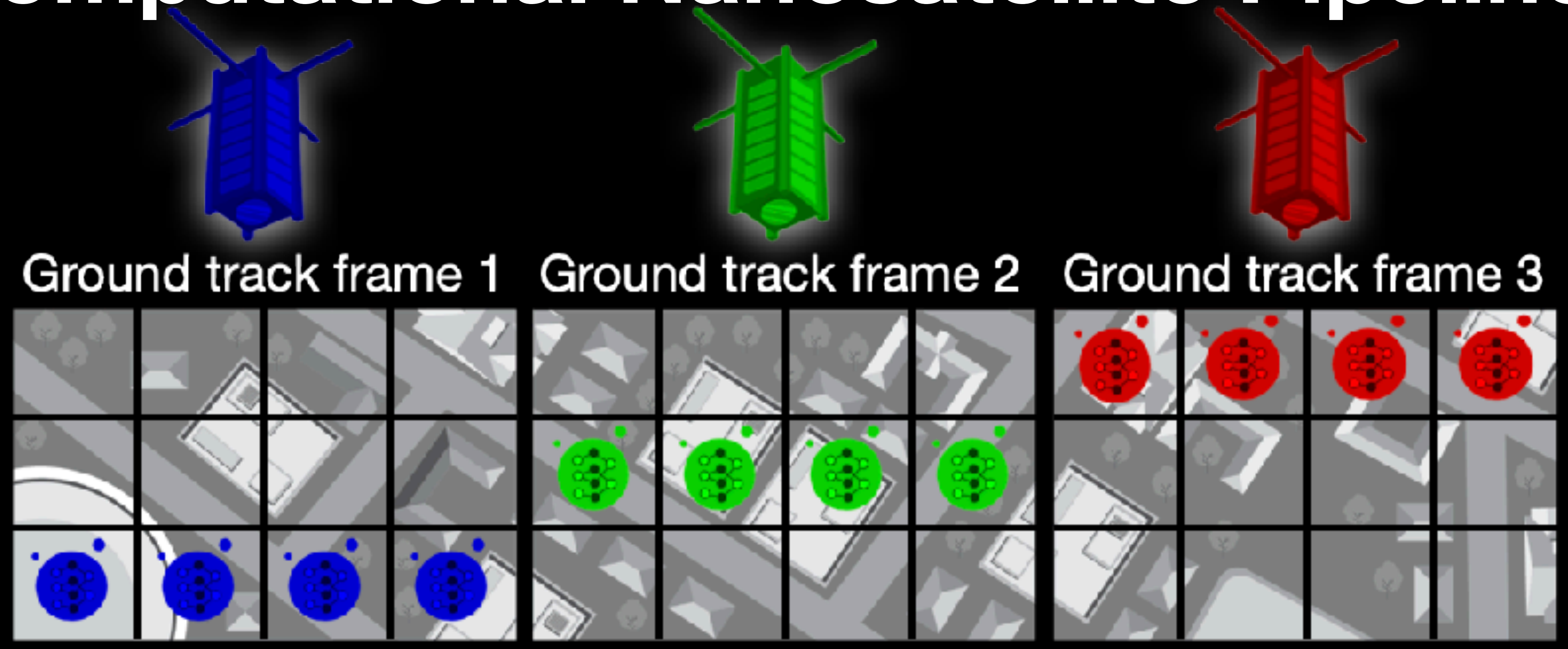
Computational Nanosatellites

Machine Learning in Space



Low (Good) GSD High (Worse) GSD

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

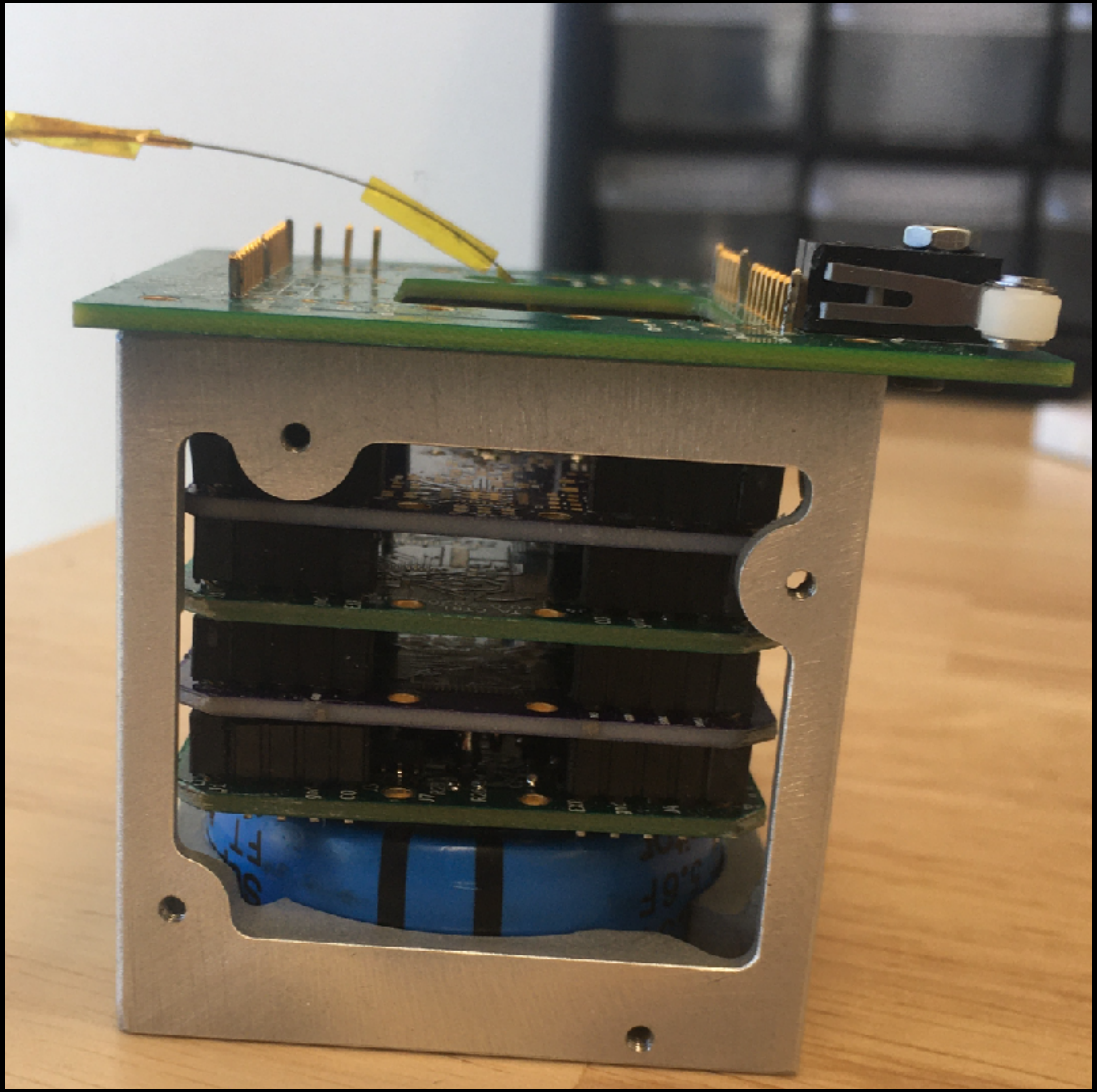
Orbital Edge Computing: Nanosatellite Constellations as a New Class of Computer System (ASPLOS'20; B Denby, B Lucia)

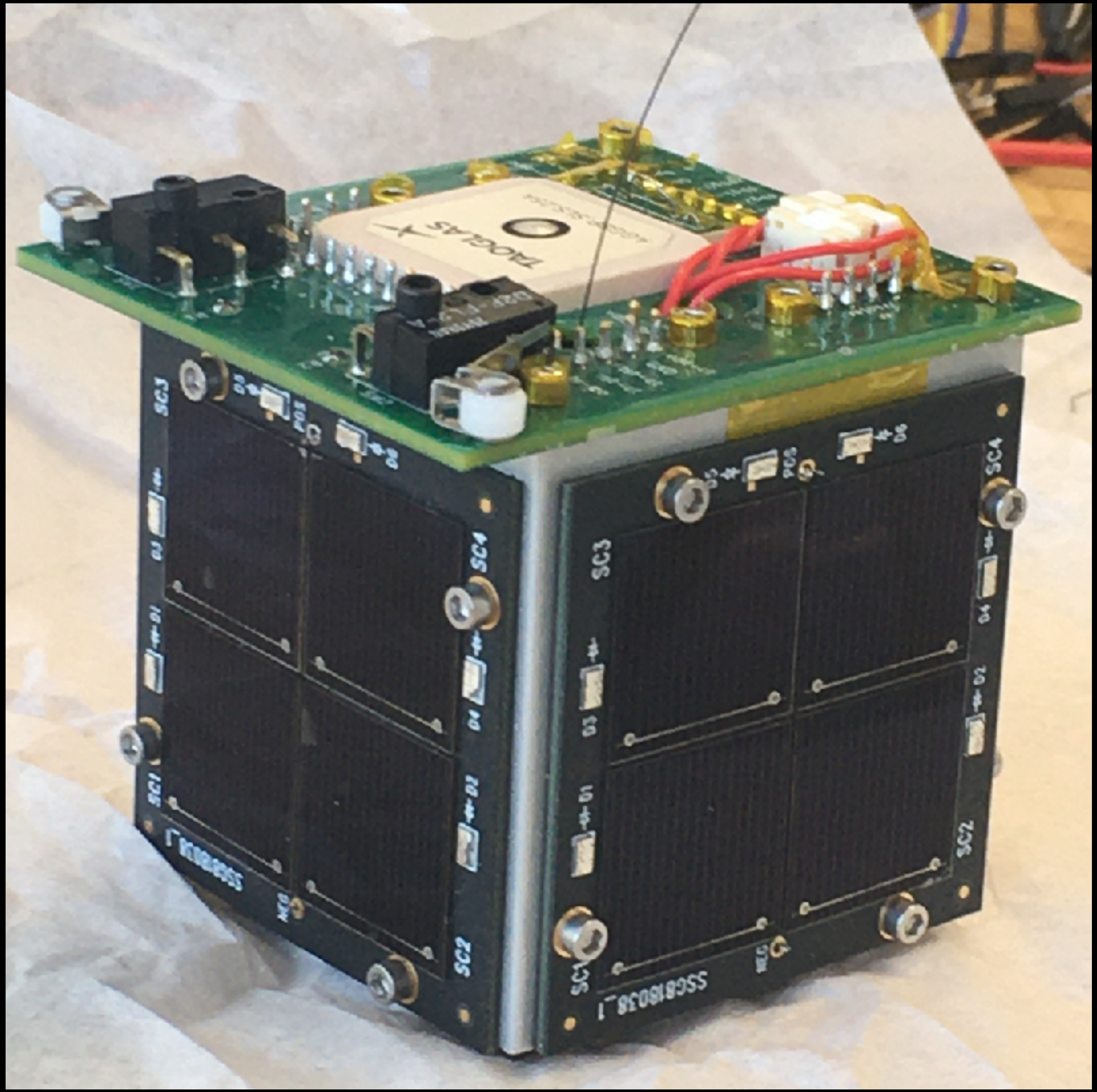
Emerging Challenges for Computational Nanosatellites

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

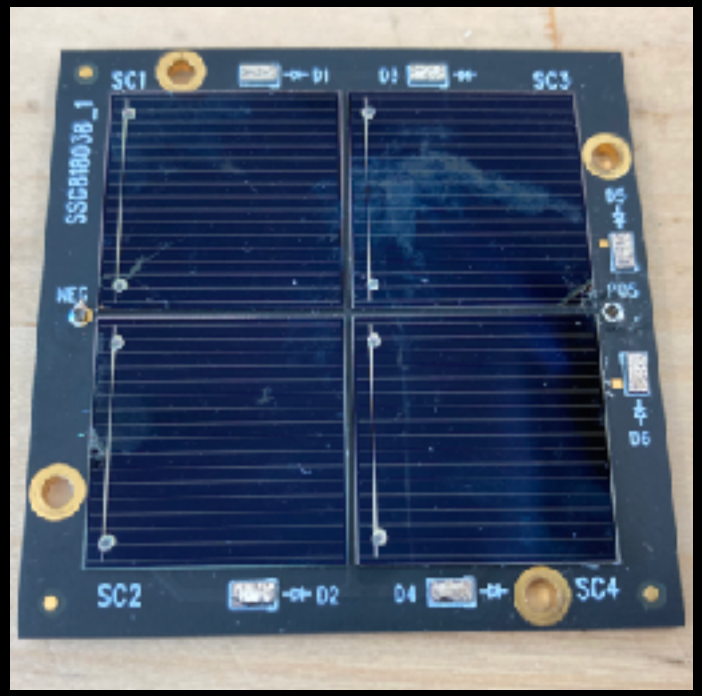
5 cm





Tartan Artibeus: Evaluation in Space

Energy System



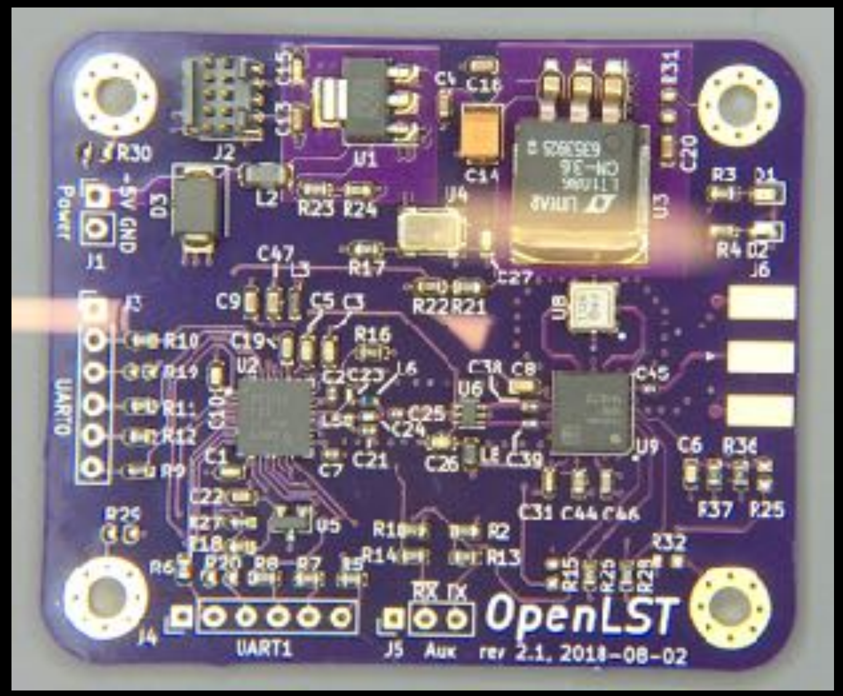
Command and Control

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

Pocketqube Radio

Computational Payload

cote

https://github.com/cmuabstract/tartan-artibeus-hw

Overview

Background and Related Work

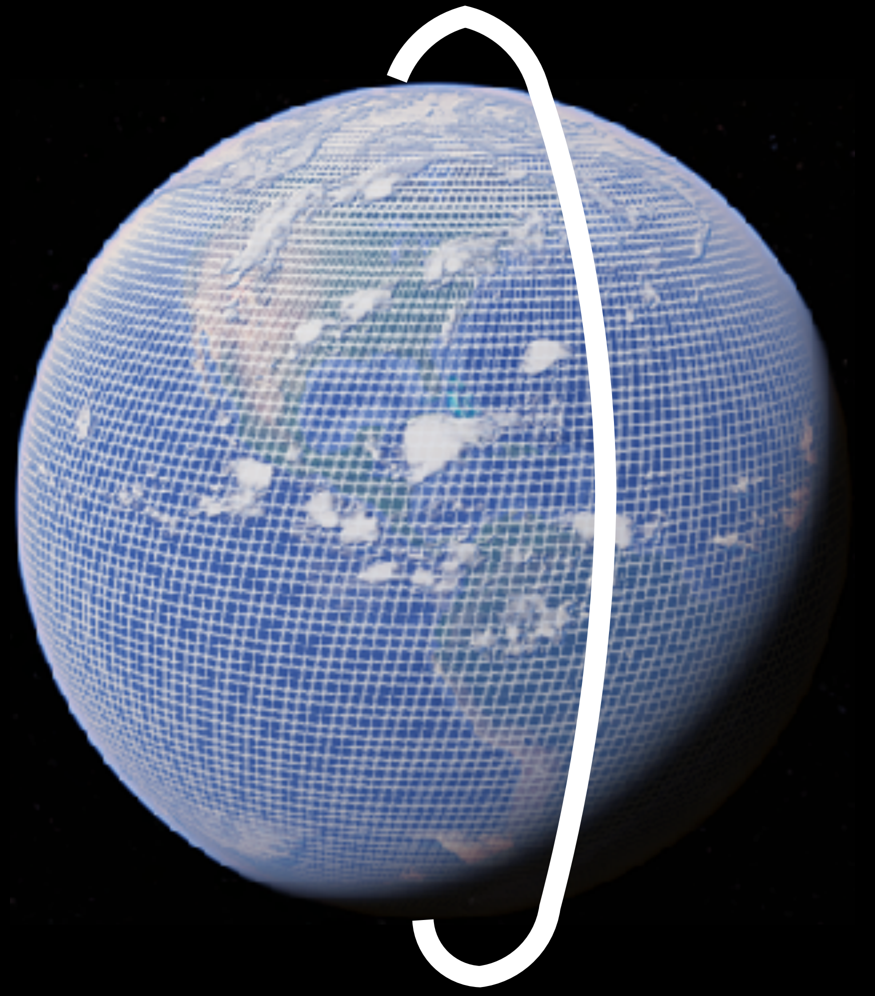
Components of Space Simulation

The `cote` Workflow

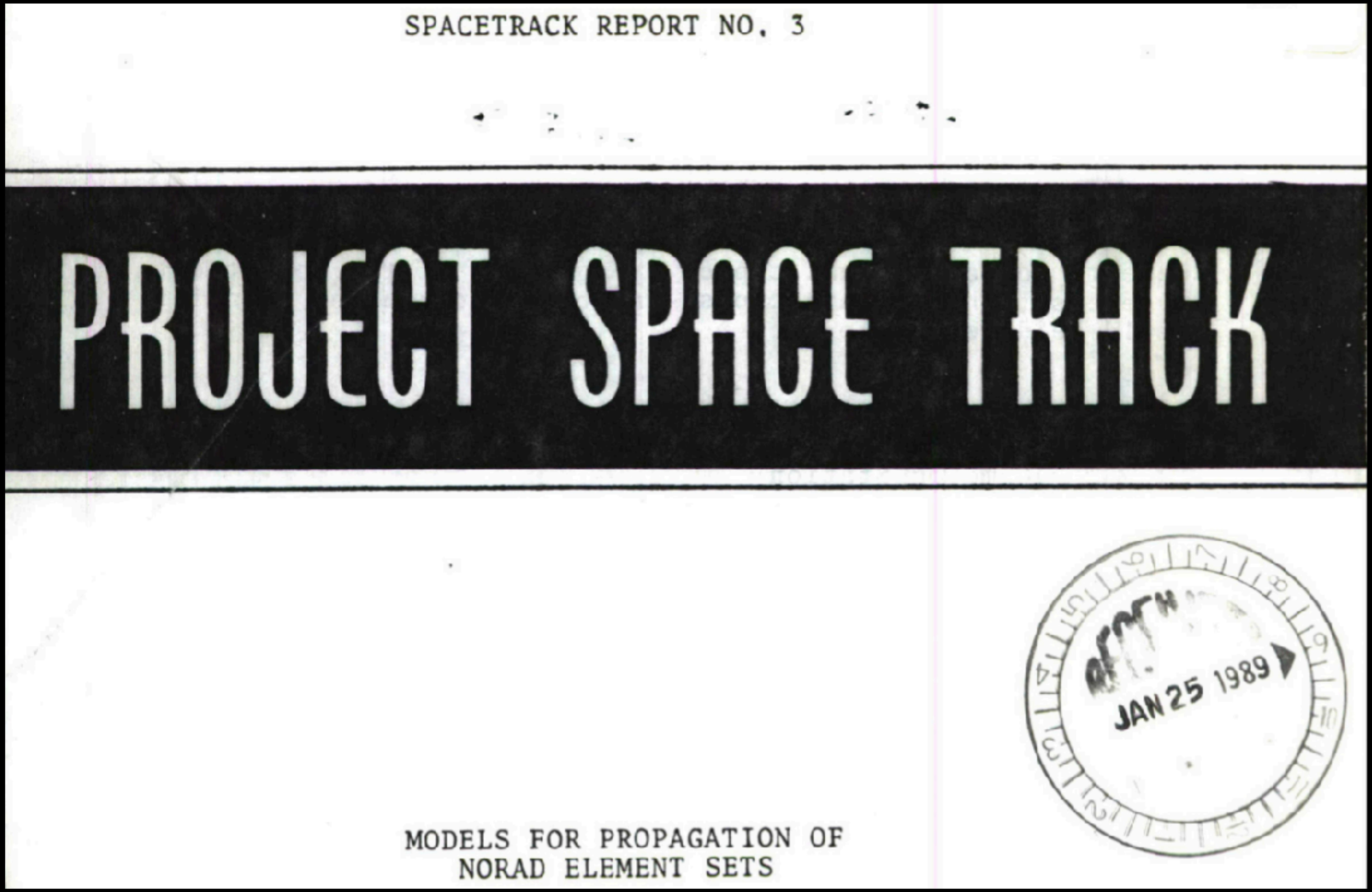
Example Programs in `cote`

Components of Space Simulation

Orbital Mechanics



Orbital Mechanics

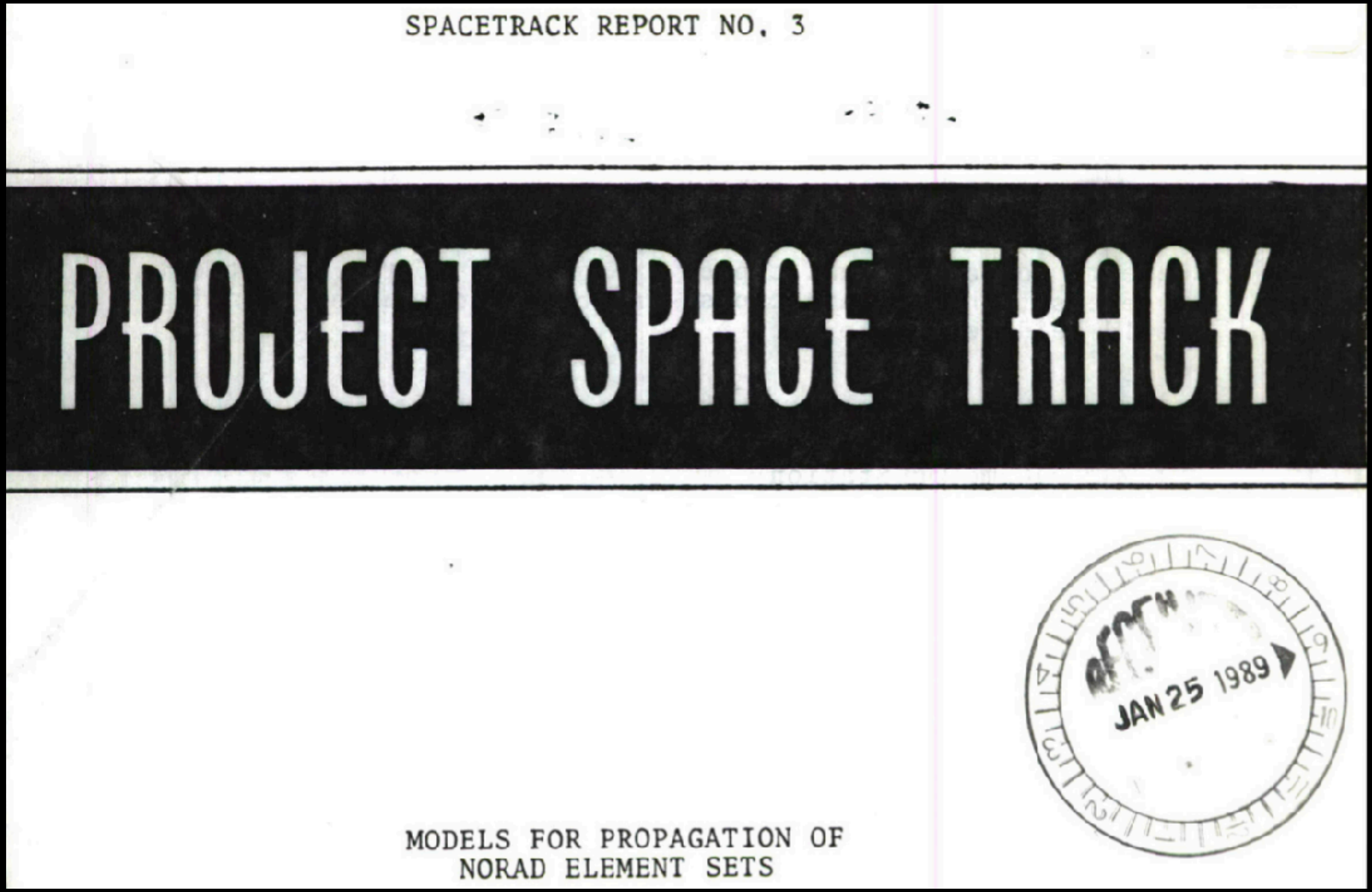

SPACETRACK REPORT NO. 3

PROJECT SPACE TRACK

JAN 25 1989

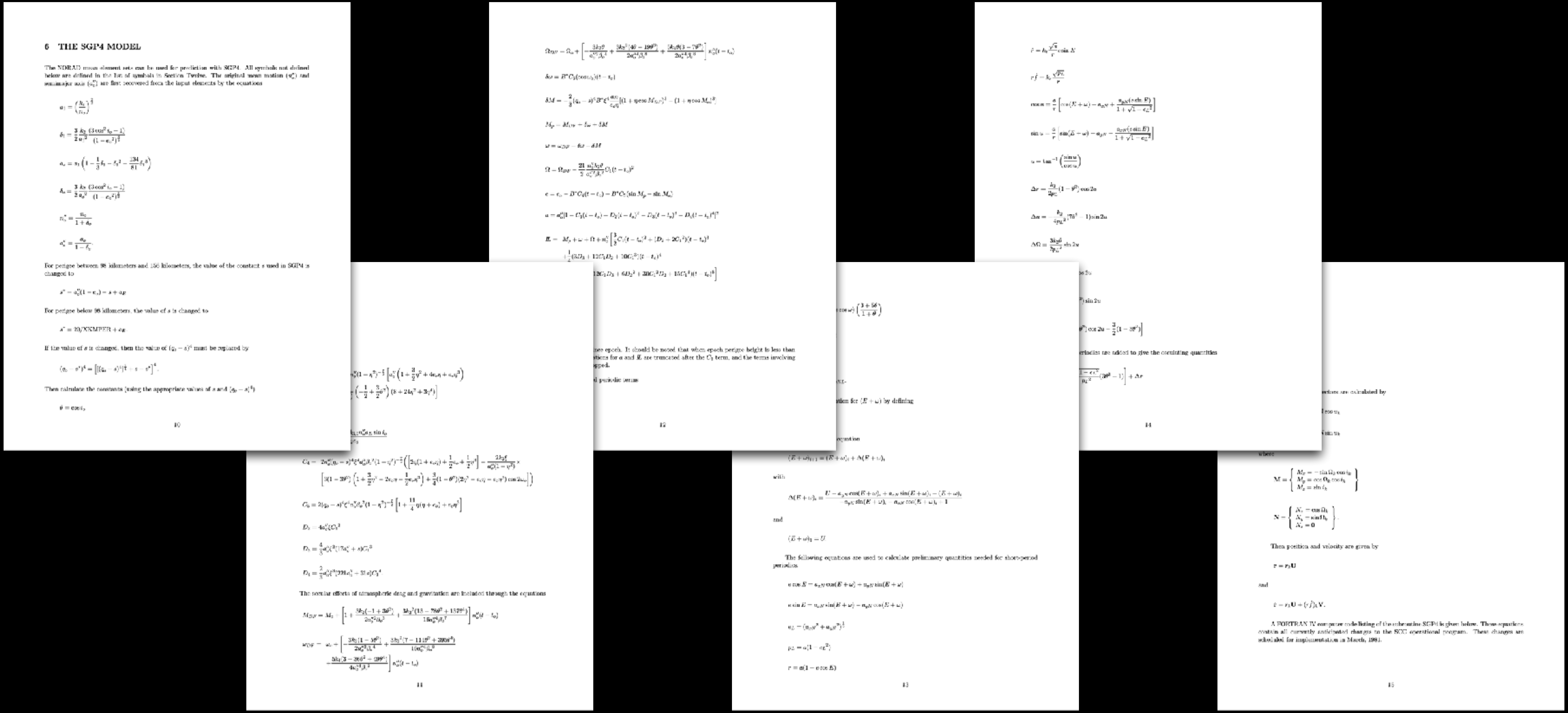
MODELS FOR PROPAGATION OF
NORAD ELEMENT SETS

Orbital Mechanics



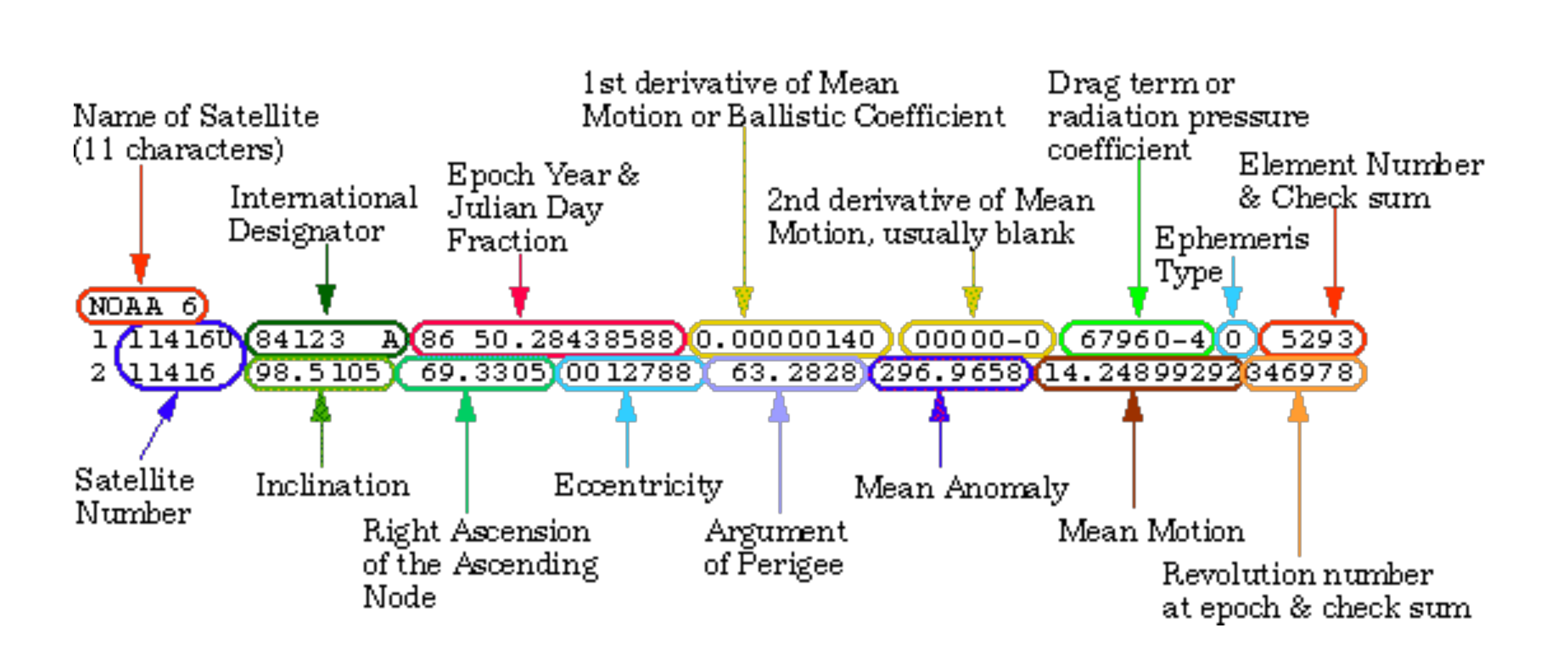
- SGP4: Simplified General Perturbations 4
  - Input: Date, Time, and Satellite TLE (two-line element set)
  - Output: Three-dimensional satellite position
- Benefits: Calculate the position of any publically-deployed satellite
- Drawbacks: Limitied utility for modeling the position of prospective satellites

SGP4 Implementation in `cote`



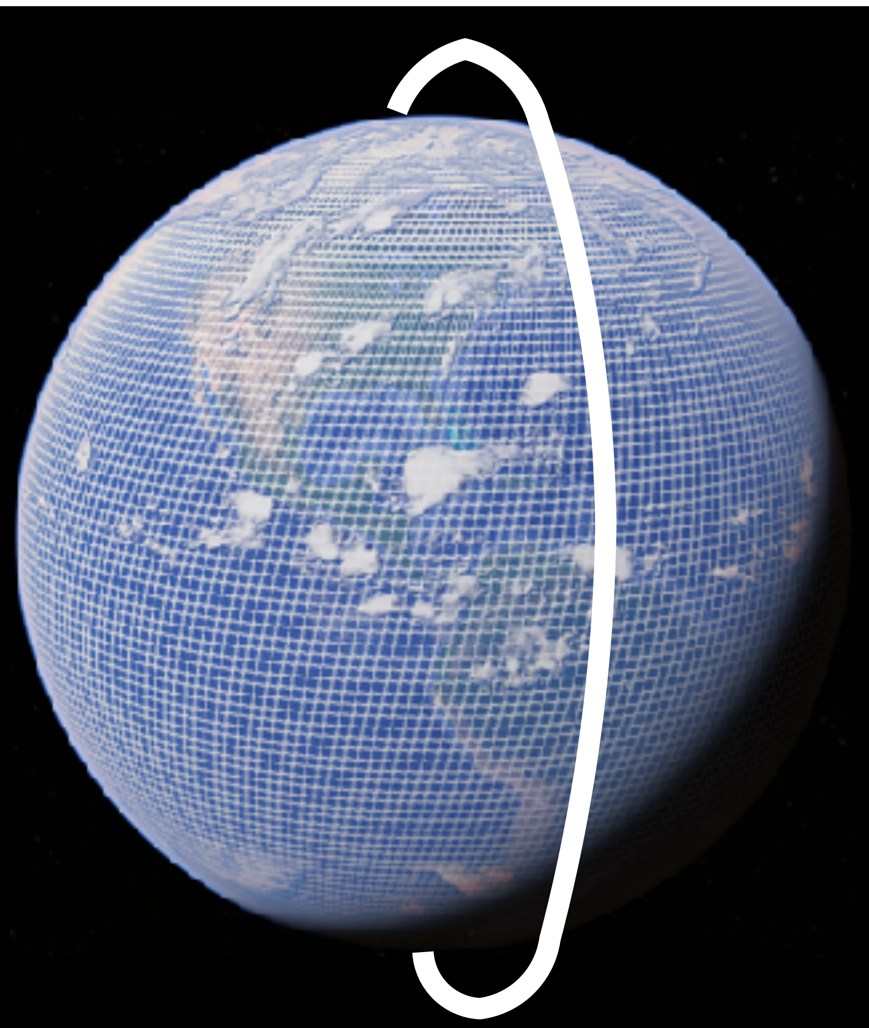
SGP4 Input: TLE (Two-line element set)

TLEs for all satellites are posted ~daily for download

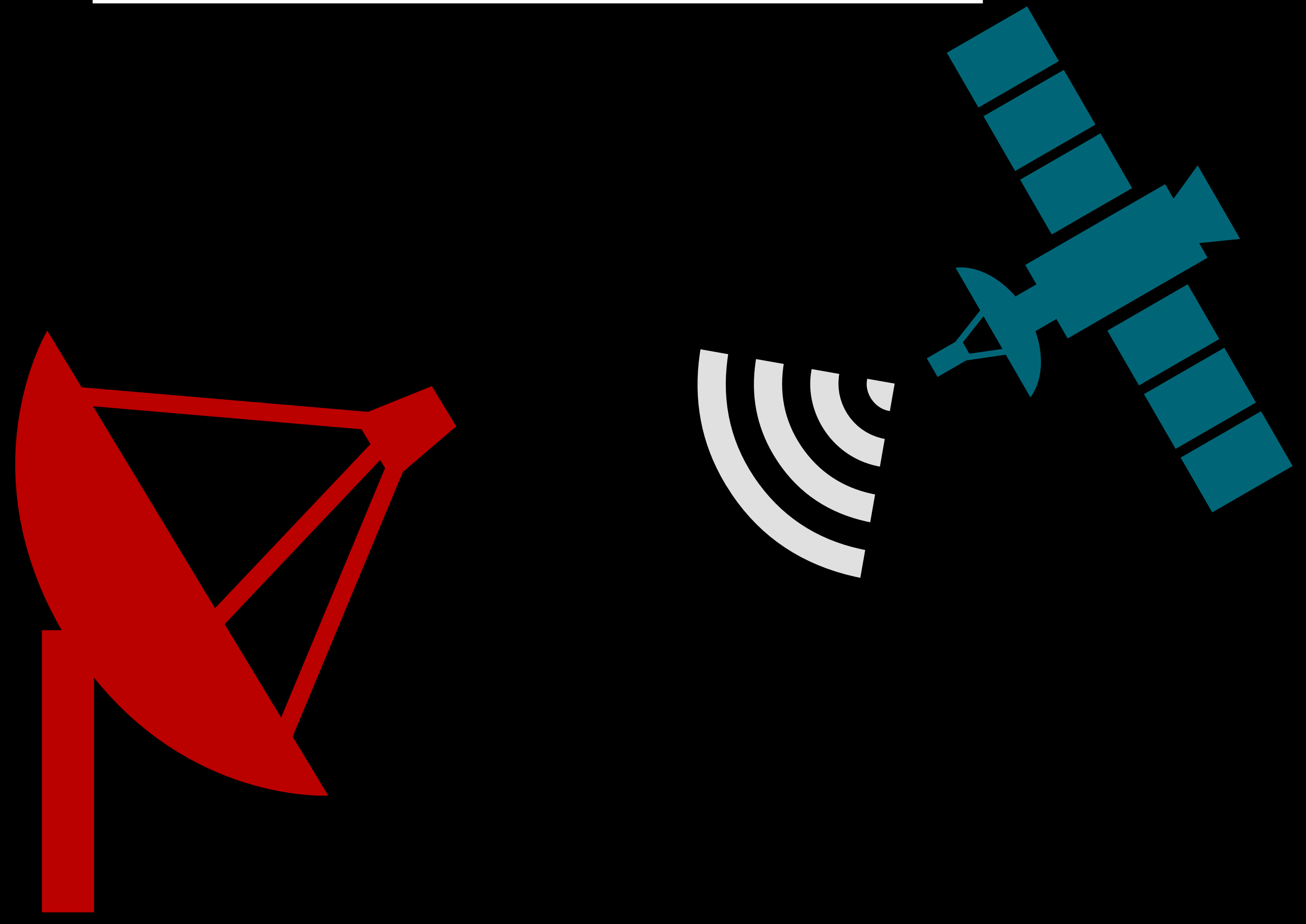


Components of Space Simulation

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.*

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

Communication Modeling

Communication Modeling

Maximum communication bit rate:

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

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 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

Received noise power N:

$$N = k T_s B$$

k: Boltzmann constant
T_s: receiver system noise temperature
B: channel bandwidth

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:

$$C = PL_lG_t\left(\frac{\lambda}{4\pi S}\right)^2 L_aG_r$$

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

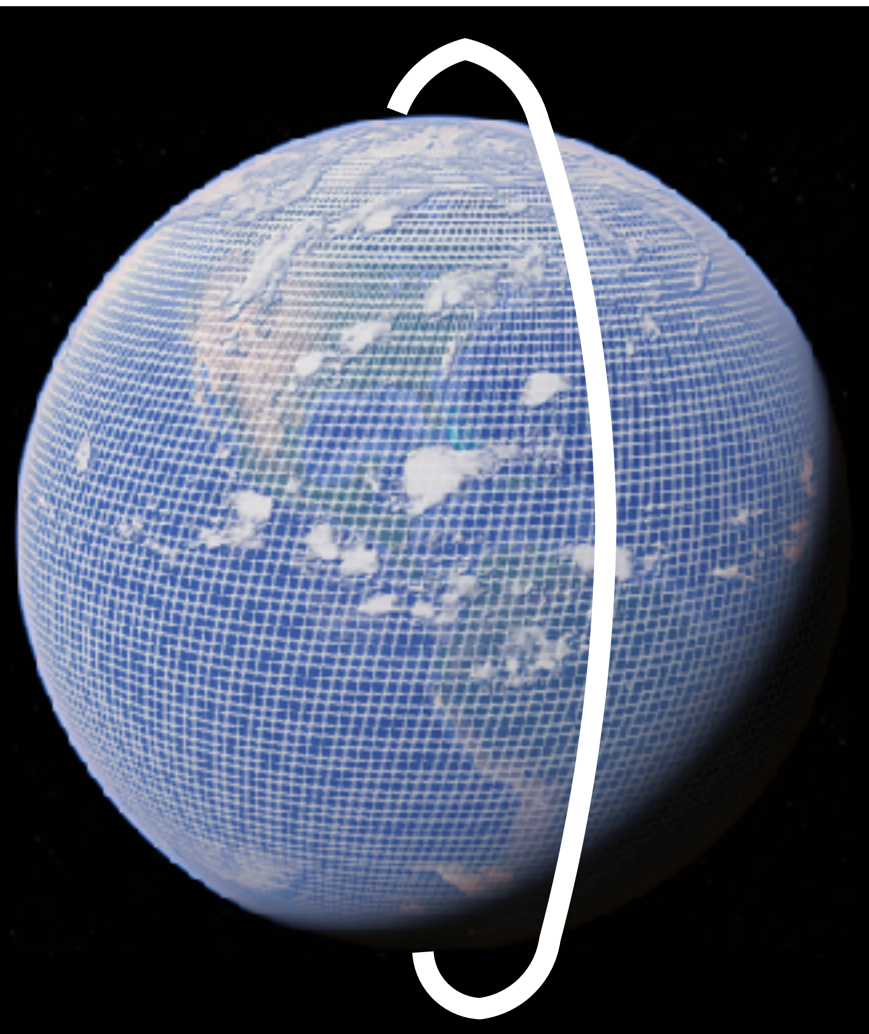
Receiver system noise temperature T_s:

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

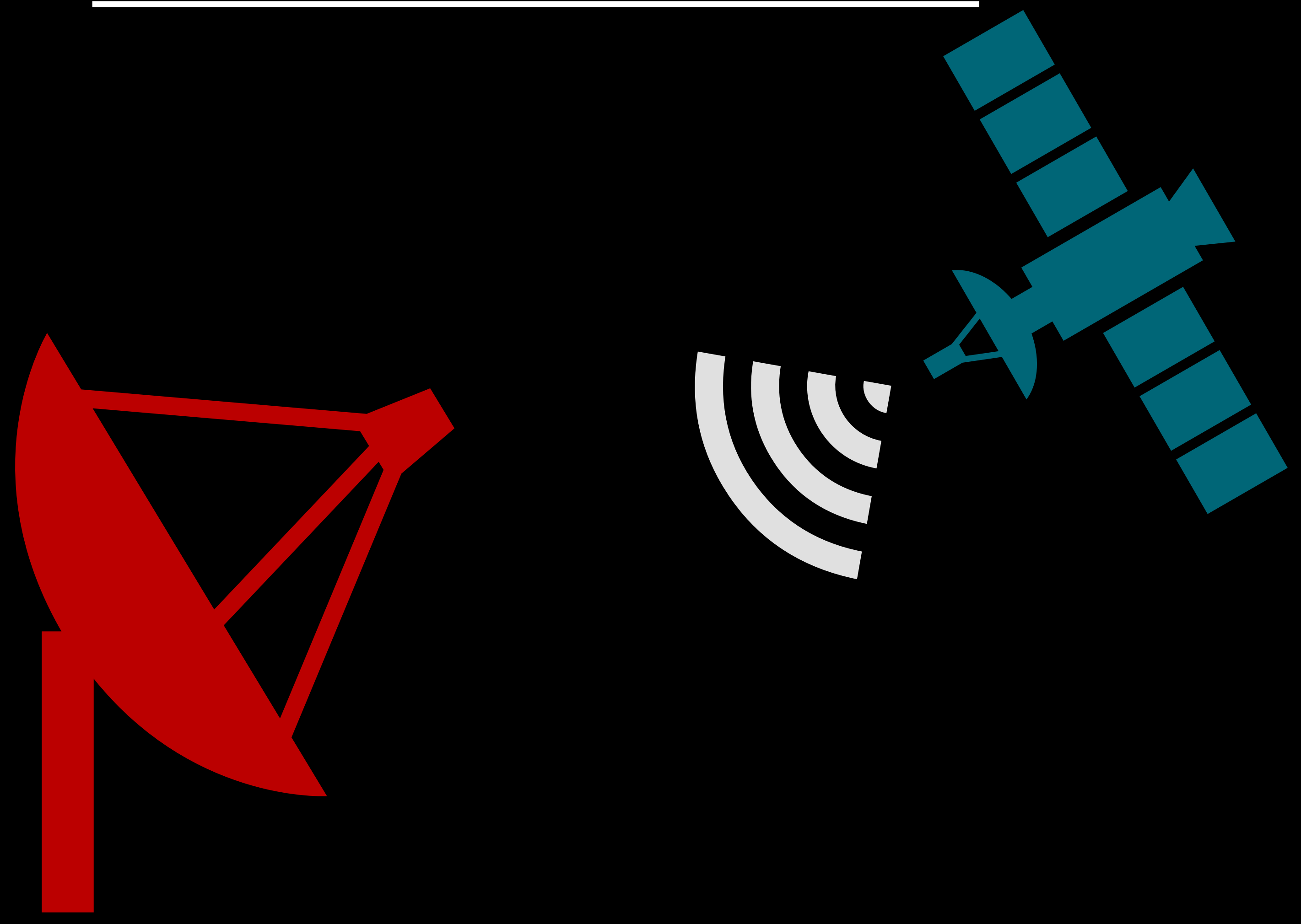
T_{ant}: RX antenna and environment factors
L_r: ANT to RX line loss
T_0: 290 K
T_r: receiver noise temperature

Components of Space Simulation

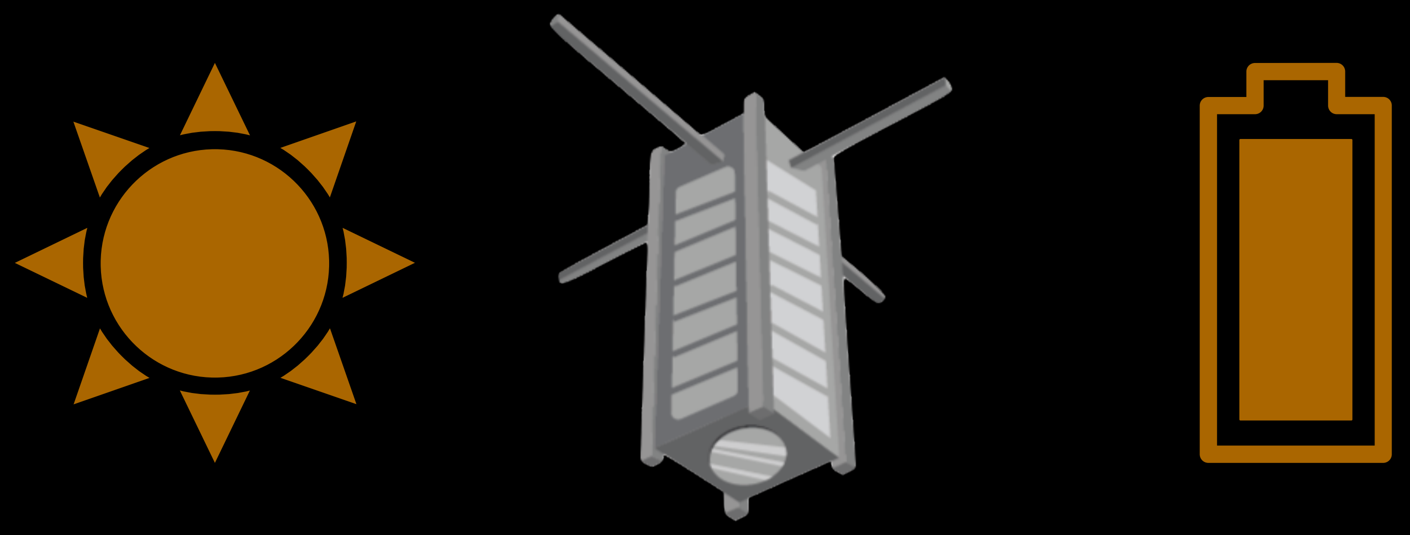
Orbital Mechanics



Communication

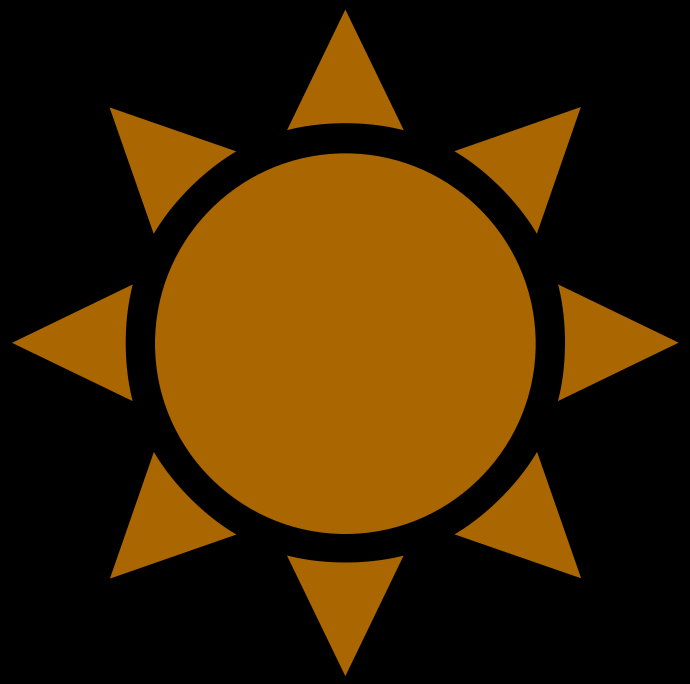


Energy and Power



Satellite Energy Systems

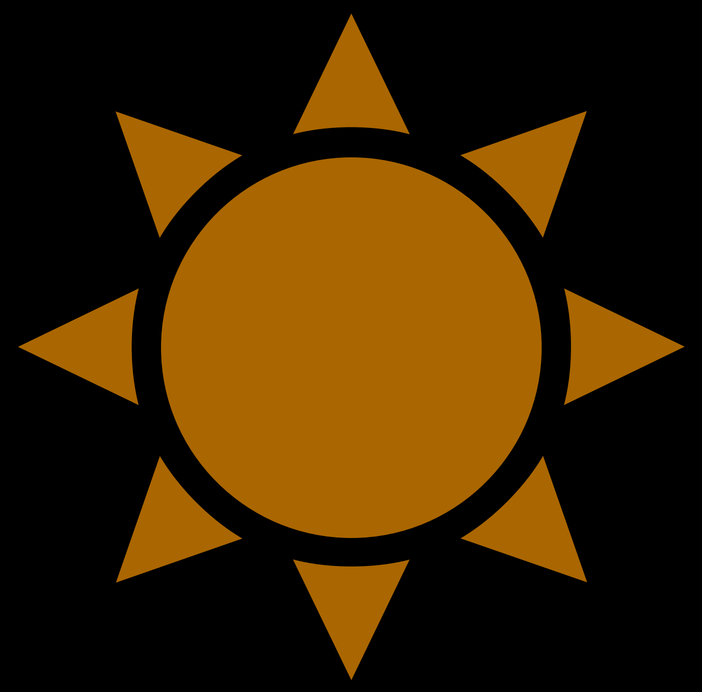
Energy Source



Irradiance
LEO: 1366 W/m²

Satellite Energy Systems

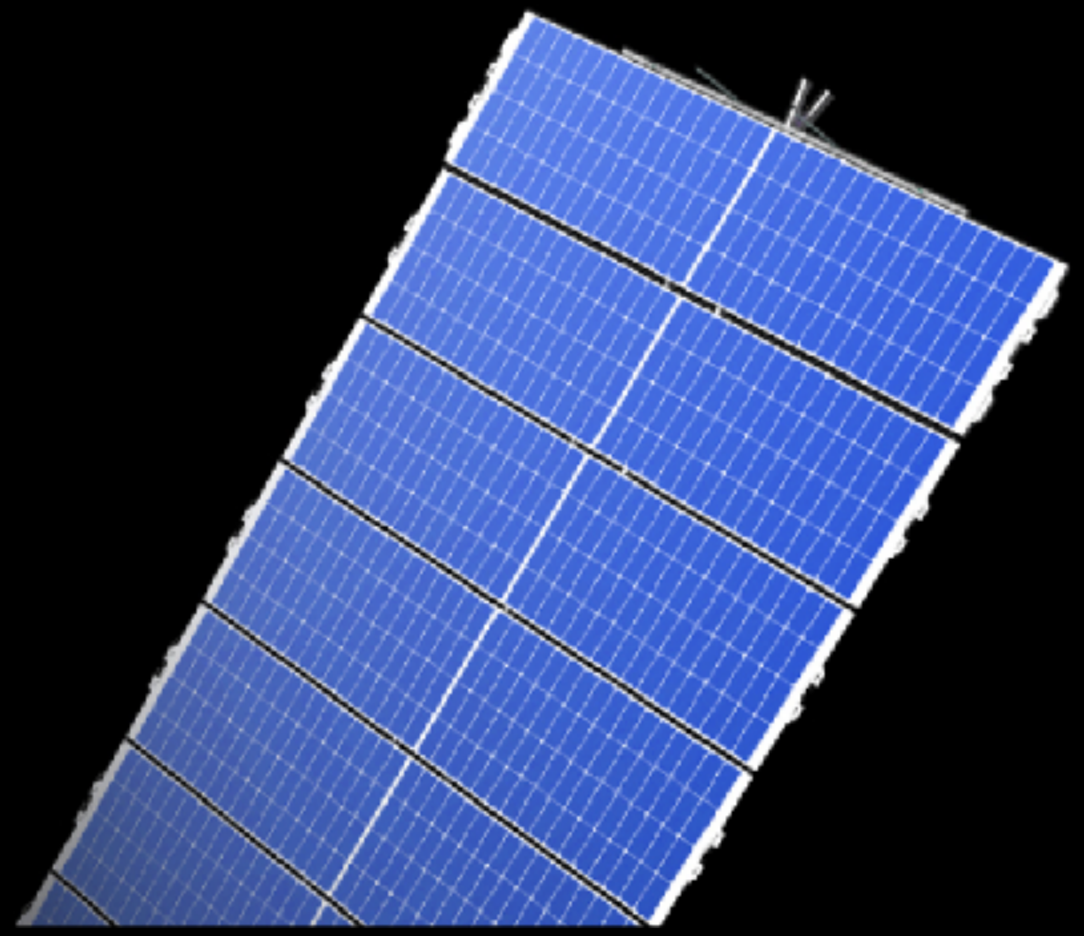
Energy Source



Irradiance

LEO: 1366 W/m²

Energy Harvester

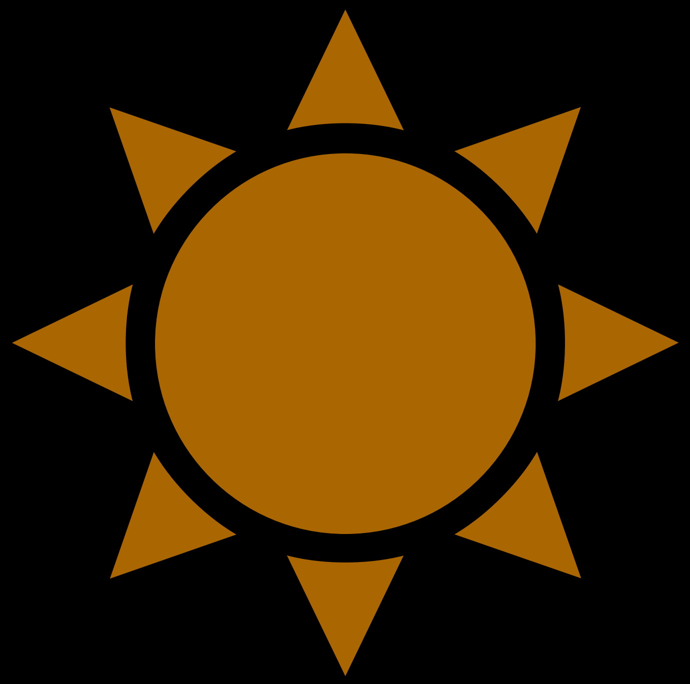


Surface Area

Efficiency

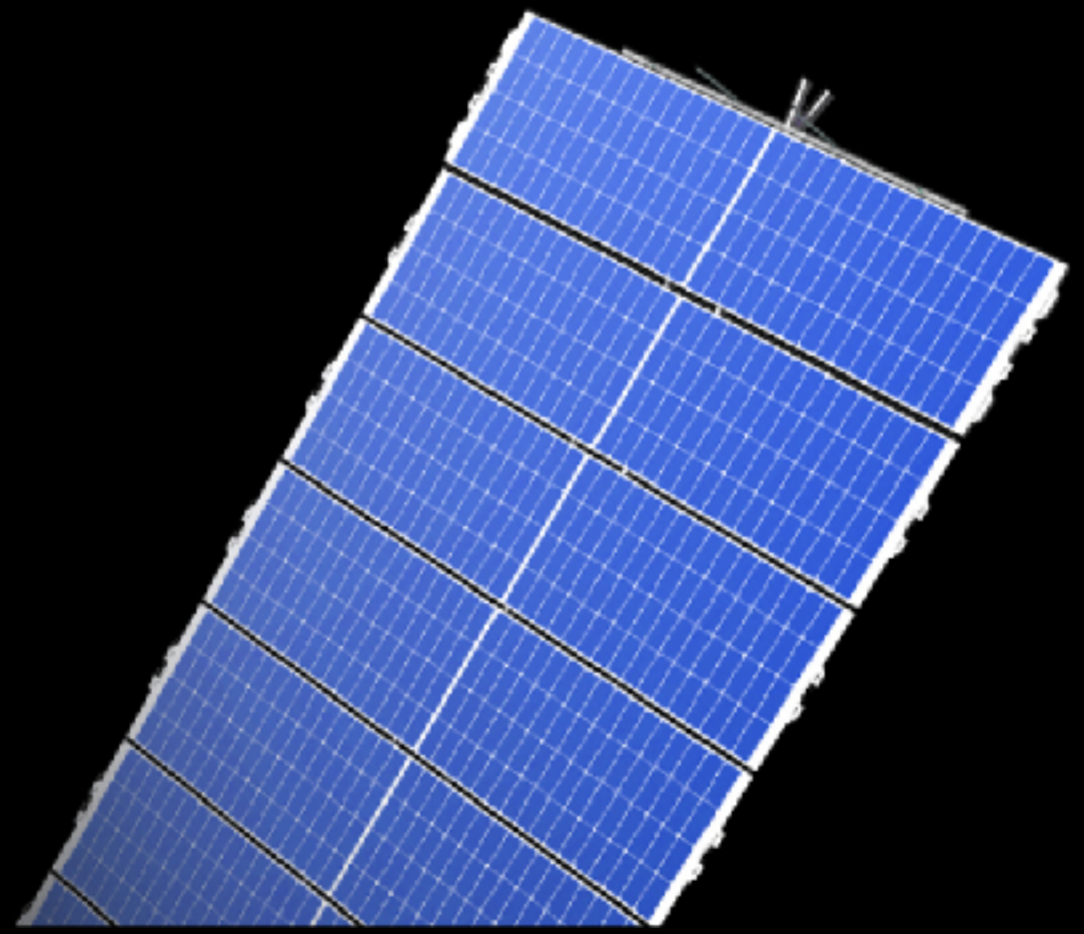
Satellite Energy Systems

Energy Source



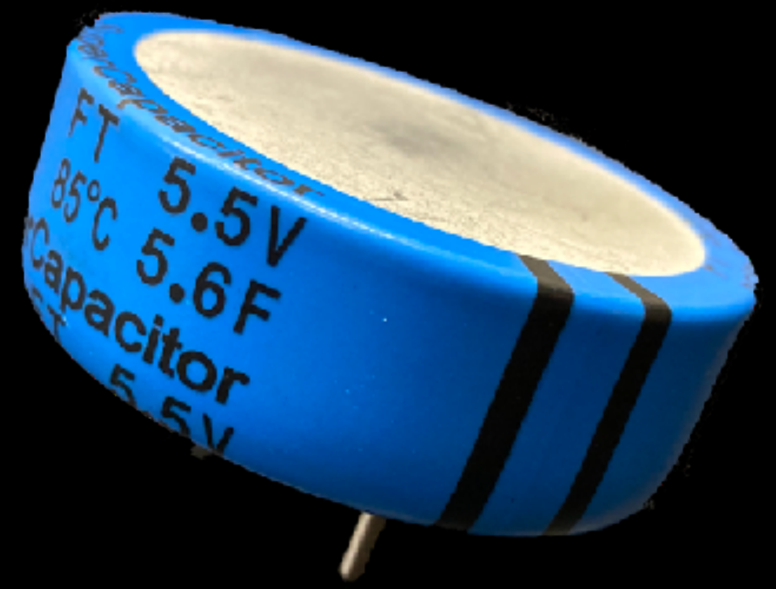
Irradiance
LEO: 1366 W/m^2

Energy Harvester



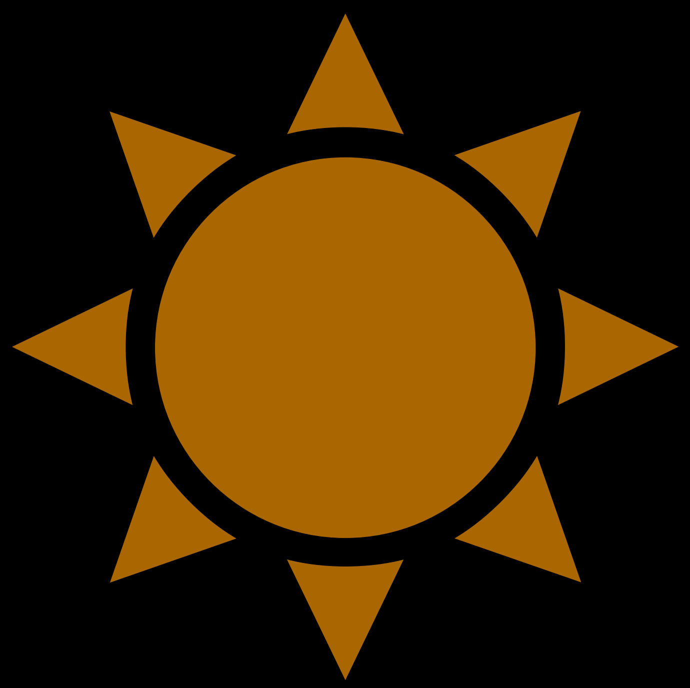
Surface Area
Efficiency

Energy Storage



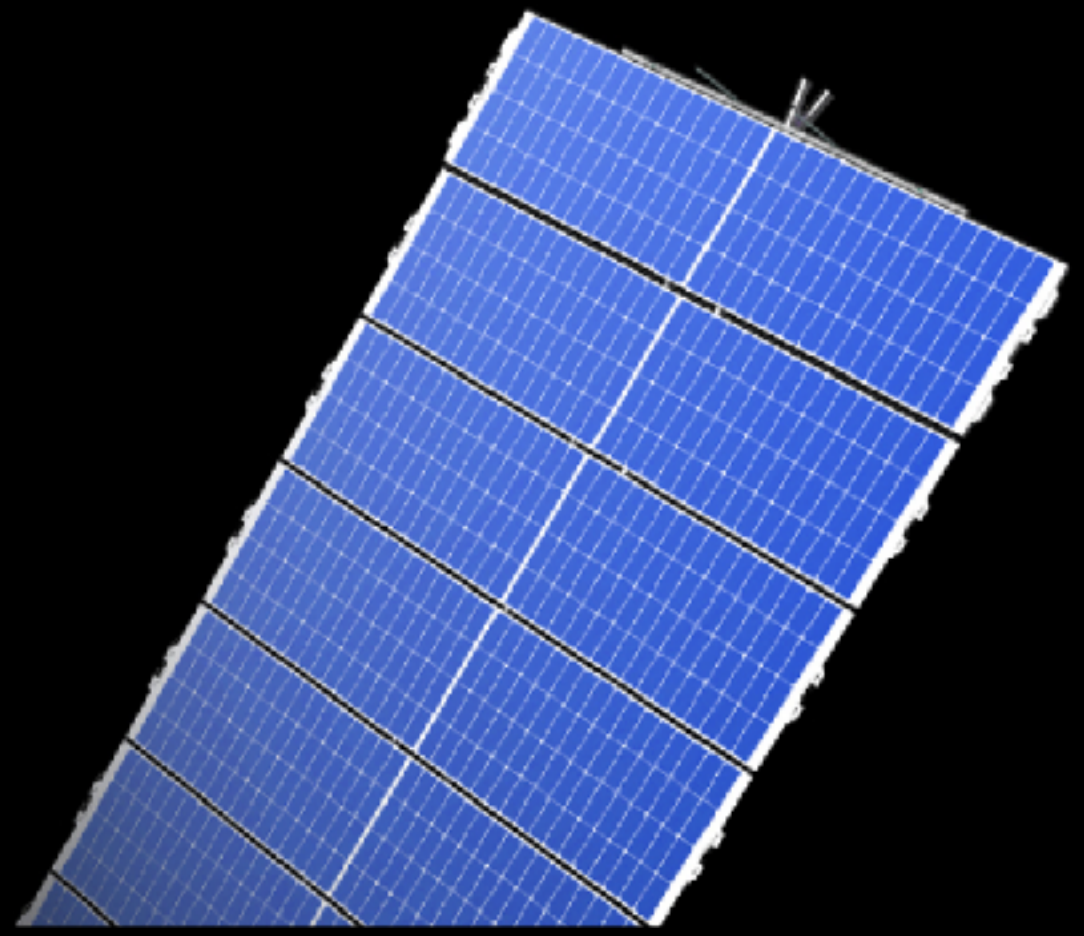
Satellite Energy Systems

Energy Source



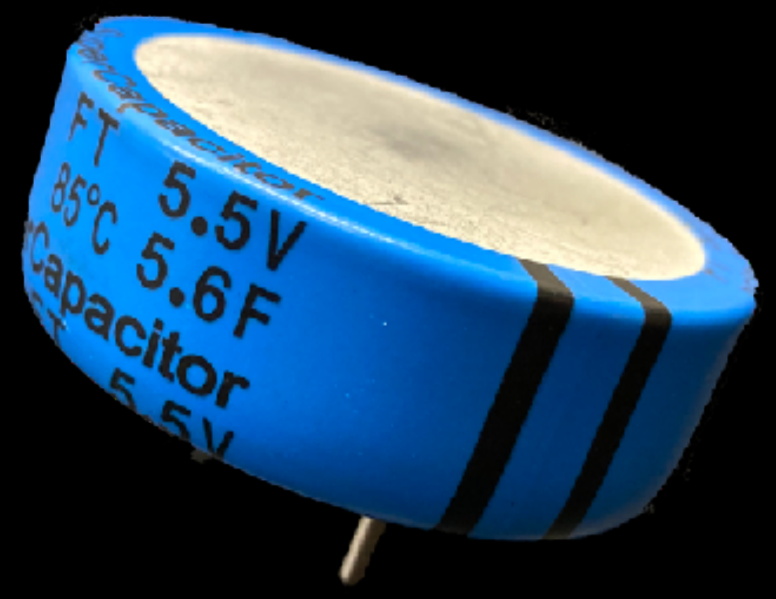
Irradiance
LEO: 1366 W/m²

Energy Harvester

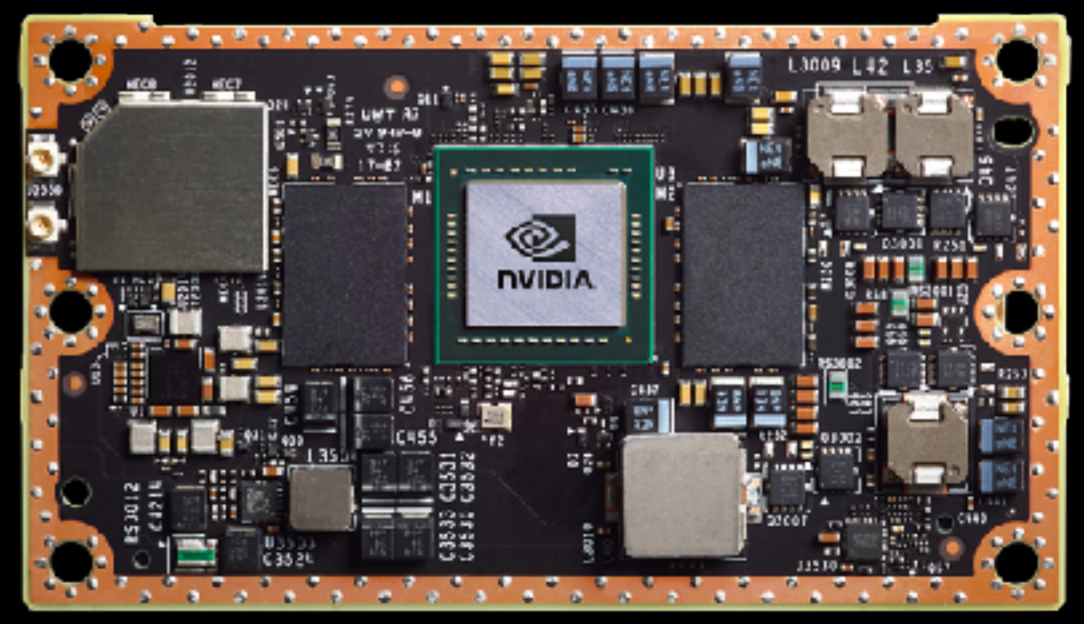


Surface Area
Efficiency

Energy Storage

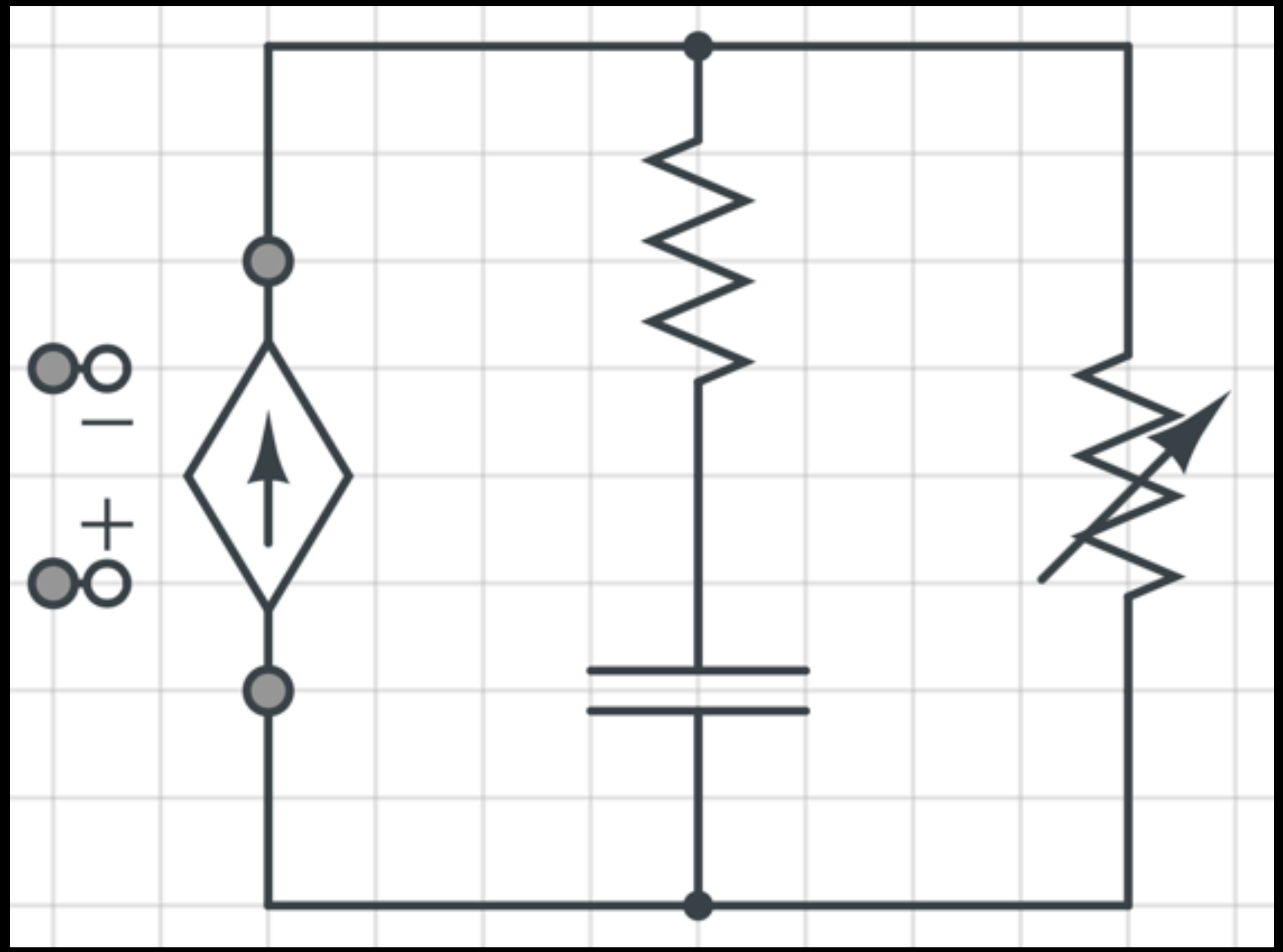


Energy Consumers



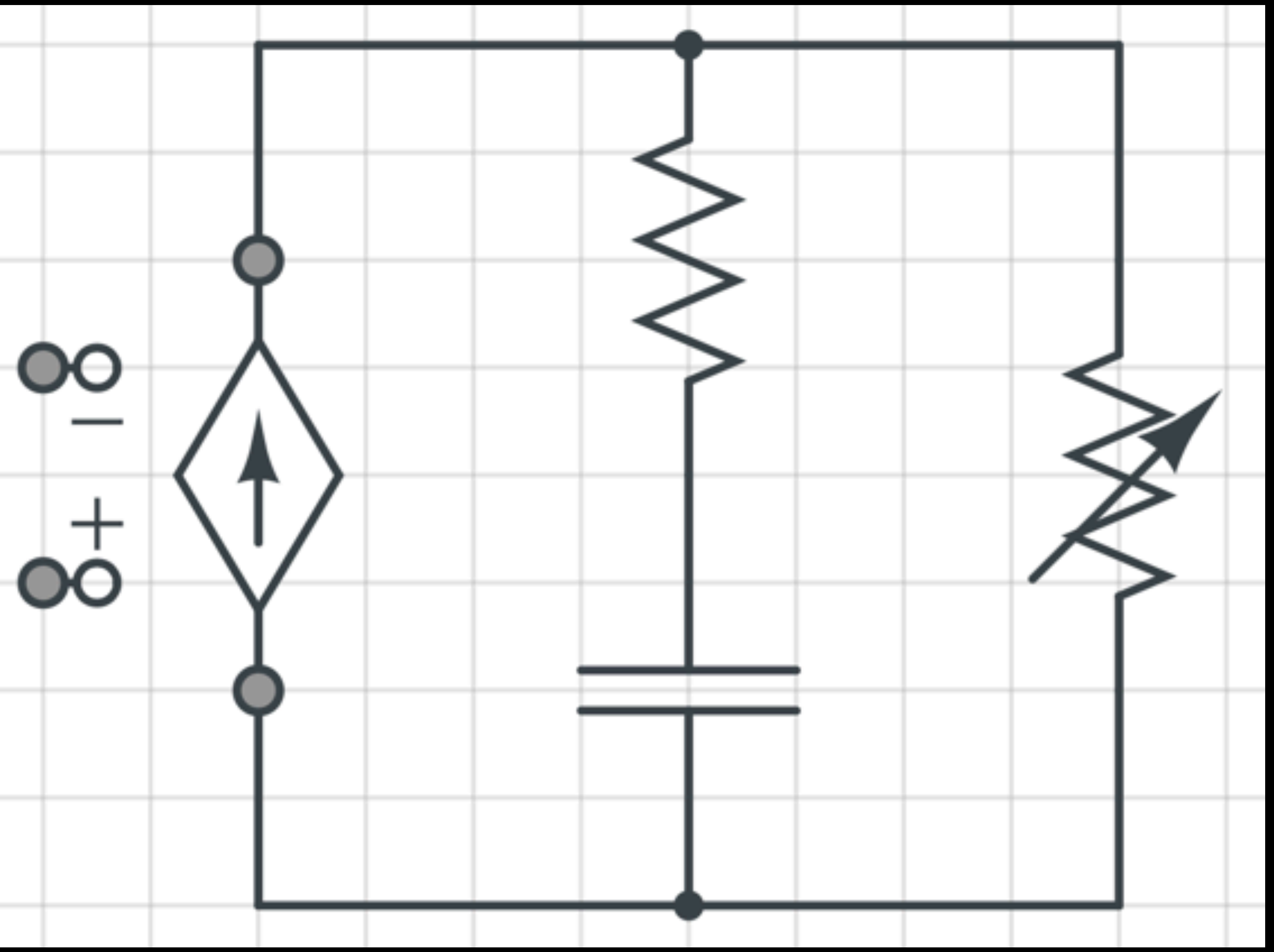
Energy Modeling

Energy Modeling



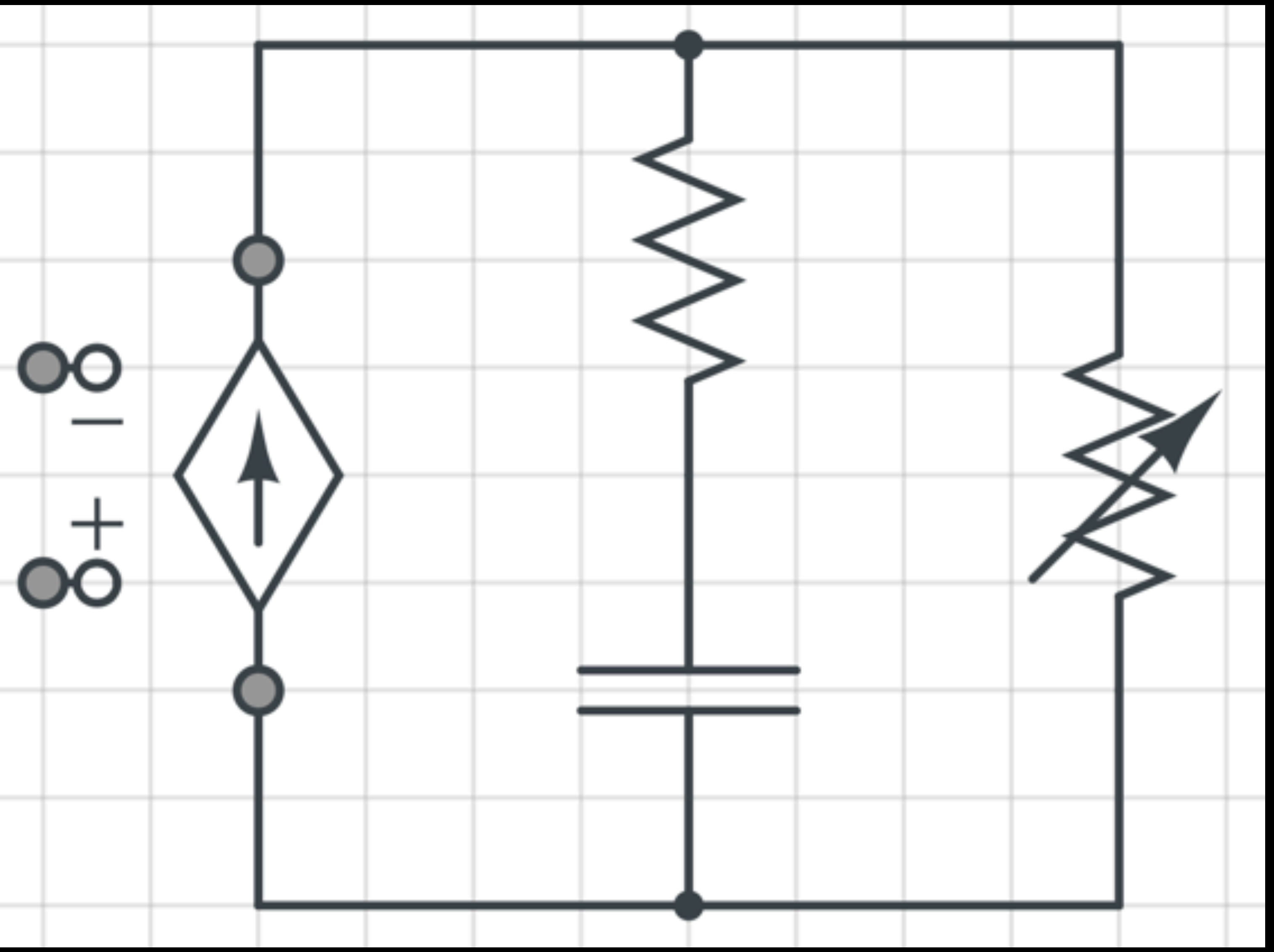
Energy Modeling

Energy harvester:
Solar cell



Energy Modeling

Energy harvester:
Solar cell

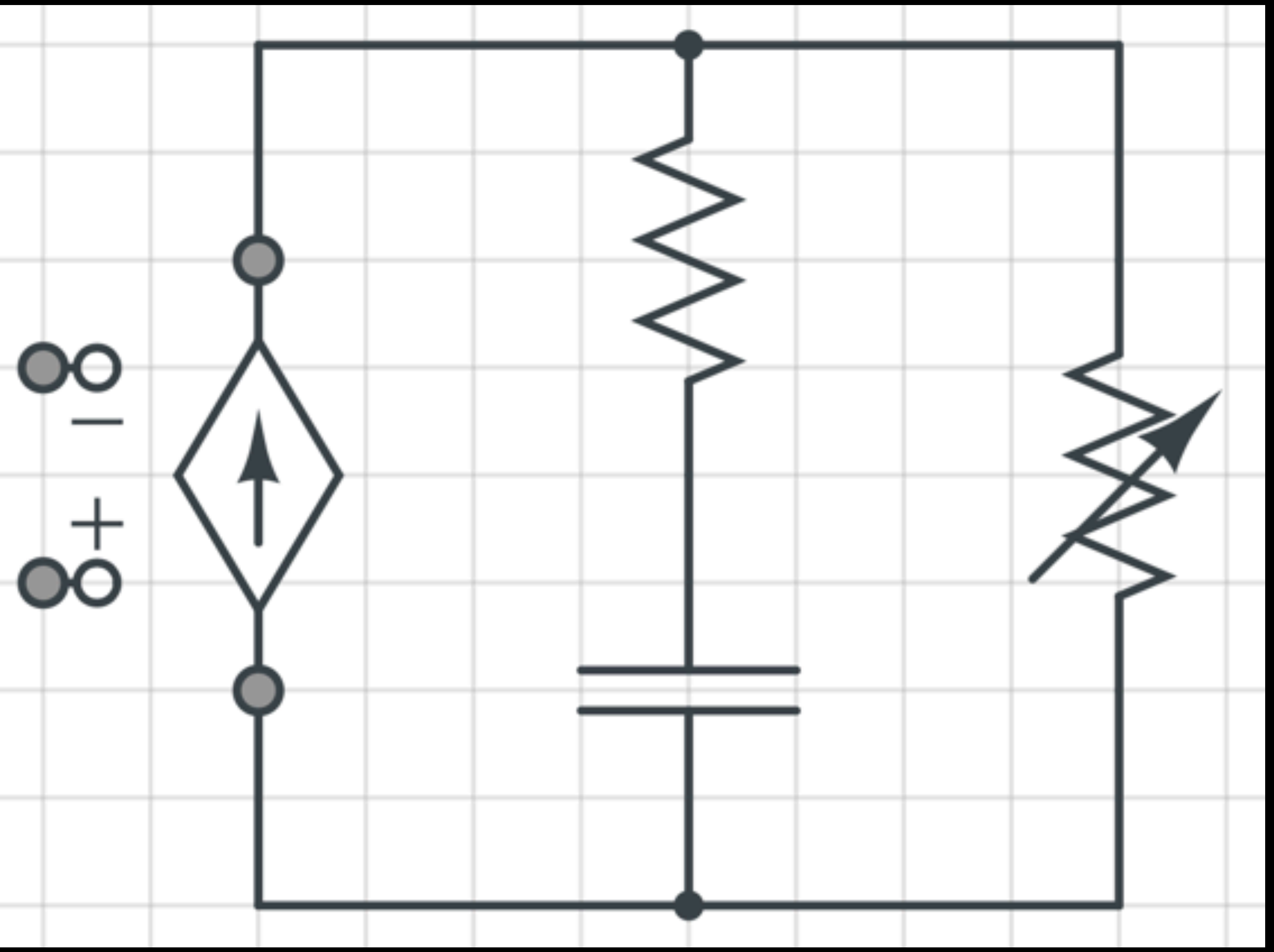


Energy storage:
Energy buffer with
Equivalent series resistance

Energy Modeling

Energy harvester:
Solar cell

Energy consumers:
Variable resistor

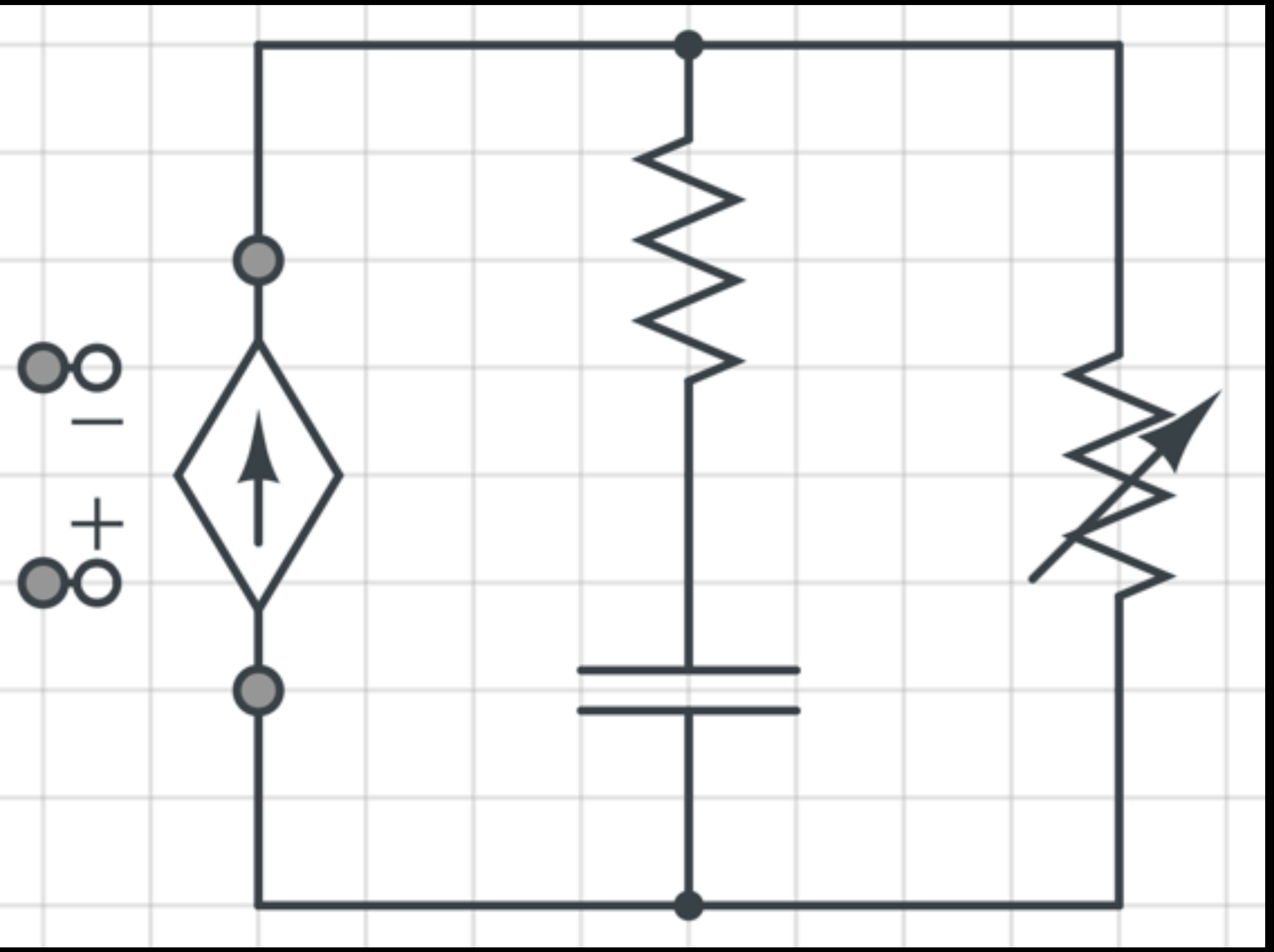


Energy storage:
Energy buffer with
Equivalent series resistance

Energy Modeling

Energy harvester:
Solar cell

Energy consumers:
Variable resistor

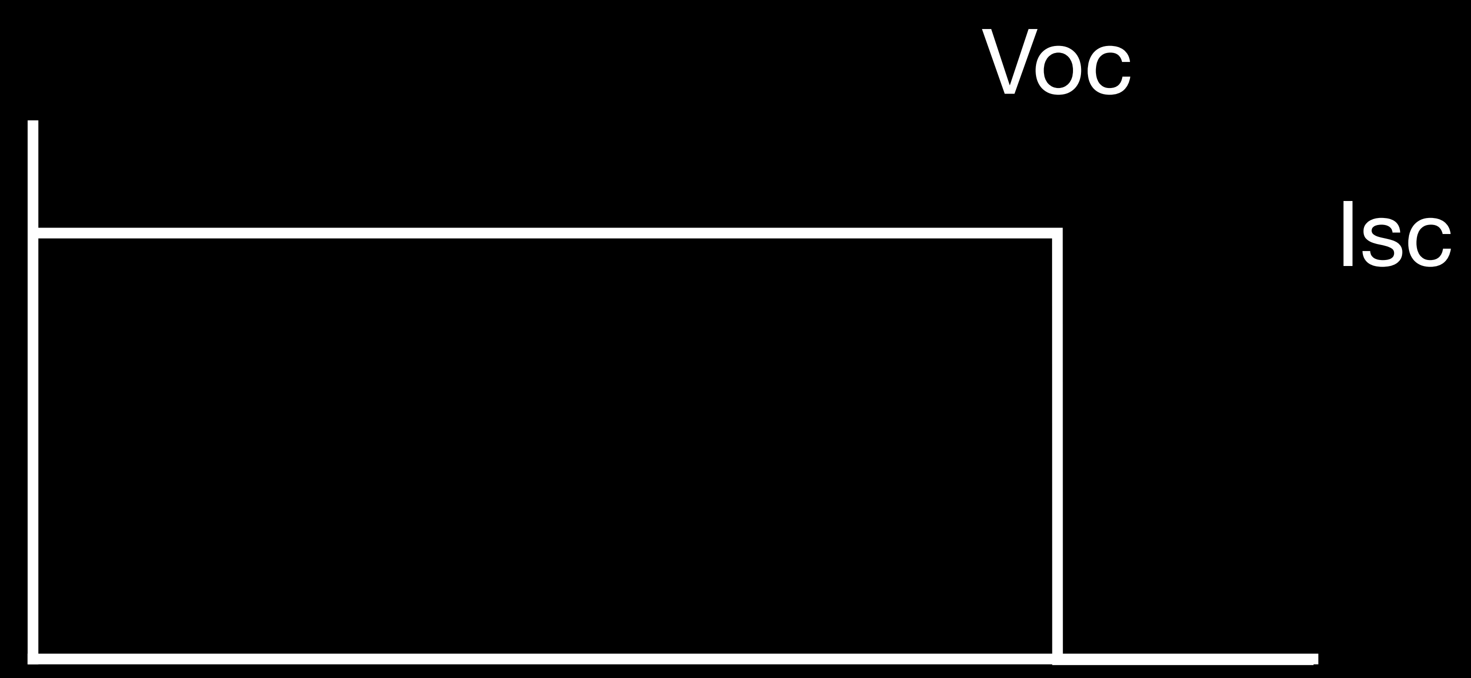


Energy storage:
Energy buffer with
Equivalent series resistance

Solar Array:

E: incident irradiance W/m^2
A: solar array surface area
e: solar array efficiency
V_{oc}: open circuit voltage
I_{sc}: short circuit current

IV Curve

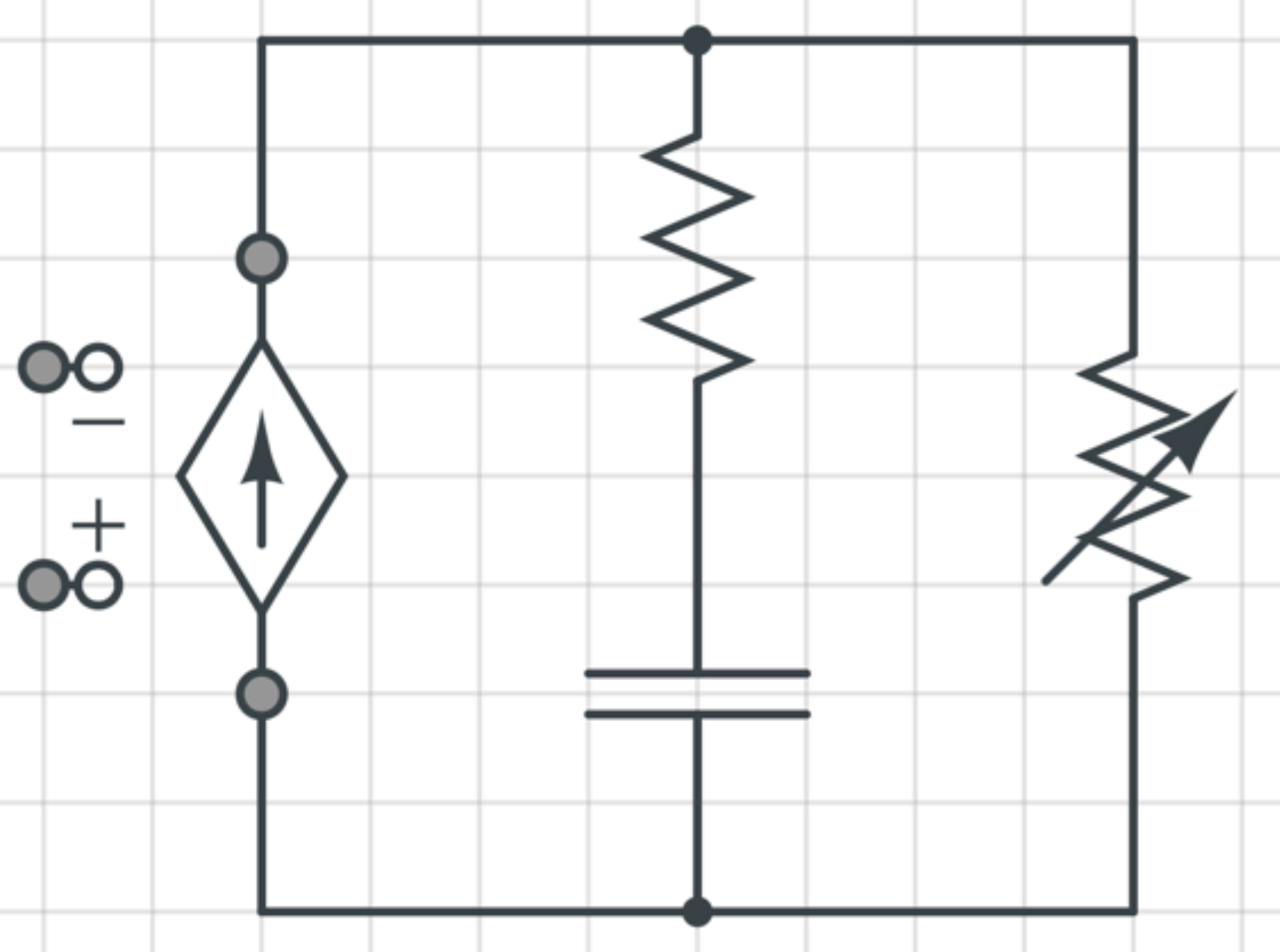


$$I_{SC} = E \cdot A \cdot e / V_{OC}$$

Energy Modeling

Energy harvester:
Solar cell

Energy consumers:
Variable resistor



Energy storage:
Energy buffer with
Equivalent series resistance

Solar Array:

E: incident irradiance W/m^2
A: solar array surface area
e: solar array efficiency
V_{oc}: open circuit voltage
I_{sc}: short circuit current

IV Curve

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

Node Voltage v(t):

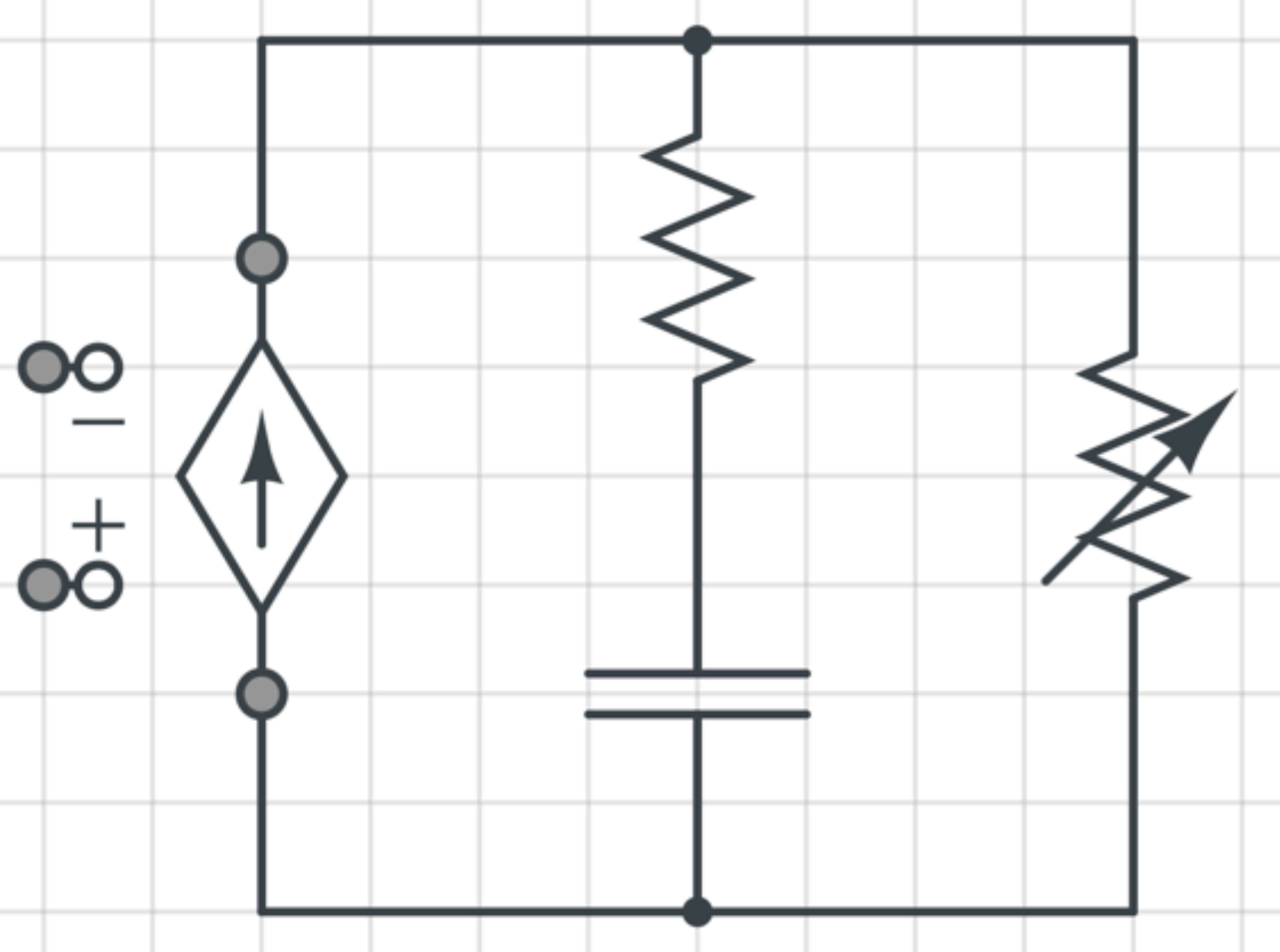
$$2v(t) = \frac{q(t)}{C} + I_S R_{ESR} + \sqrt{\left(\frac{q(t)}{C} + I_S R_{ESR}\right)^2 - 4 P_{MODE} R_{ESR}}$$

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

Energy Modeling

Energy harvester: Solar cell

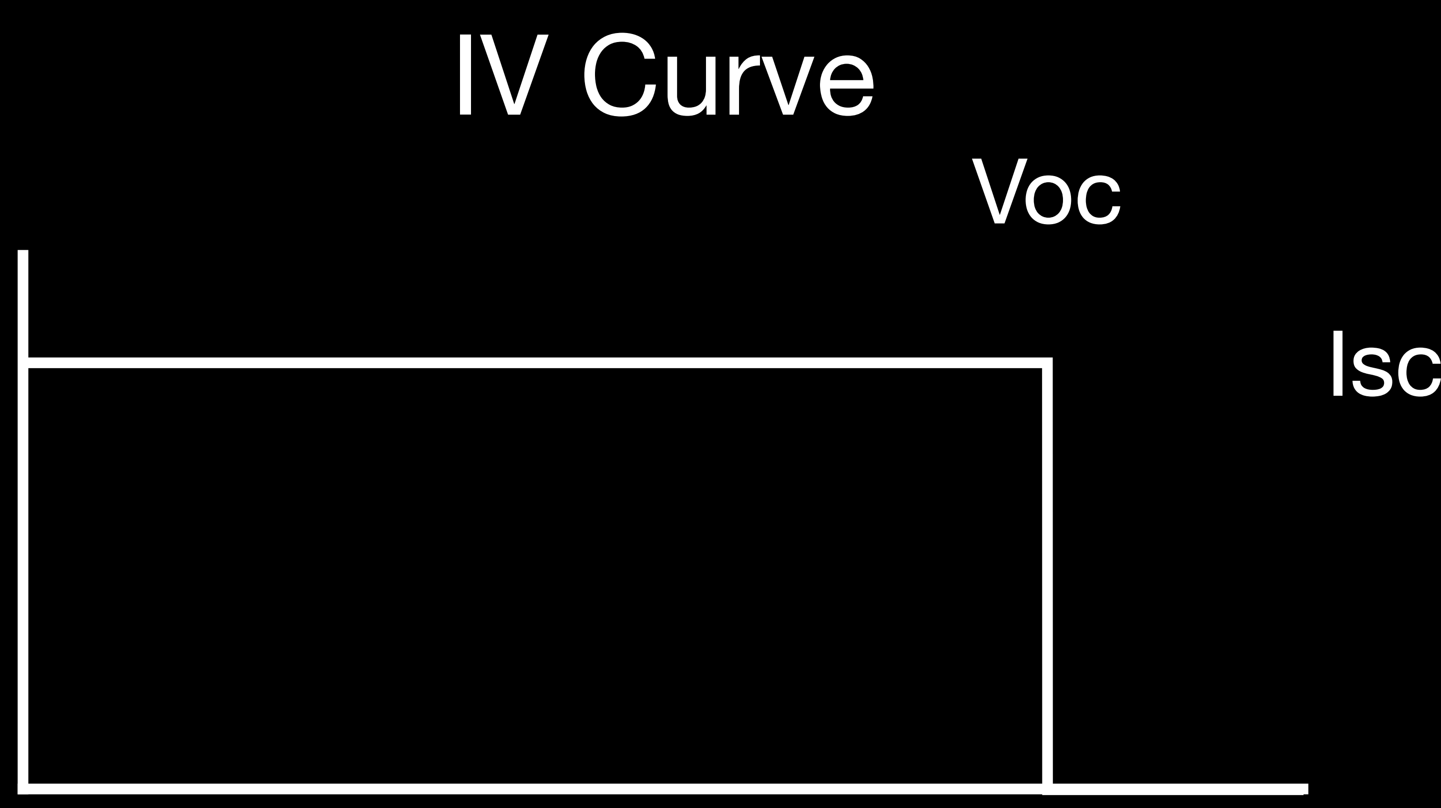
Energy consumers: Variable resistor



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



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

Node Voltage v(t):

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

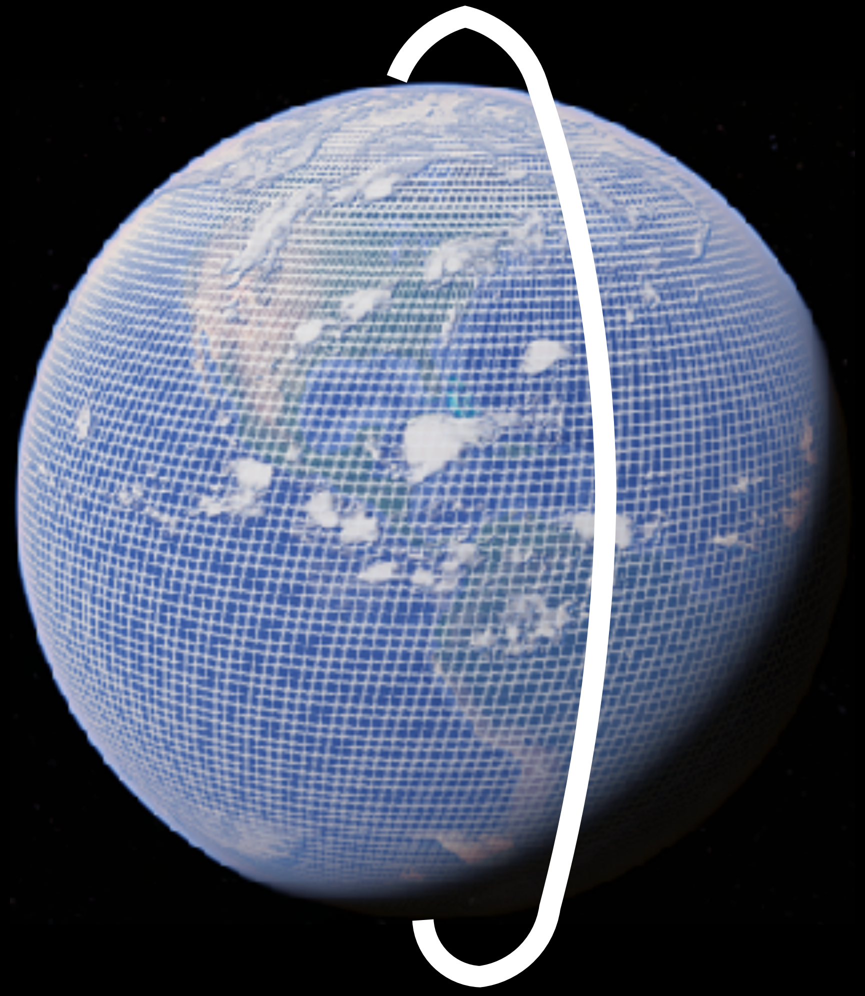
q(t): buffer charge
C: buffer capacitance
Is: solar array current
R_{ESR}: capacitor ESR
P_{MODE}: peripheral power draw

Energy consumers:

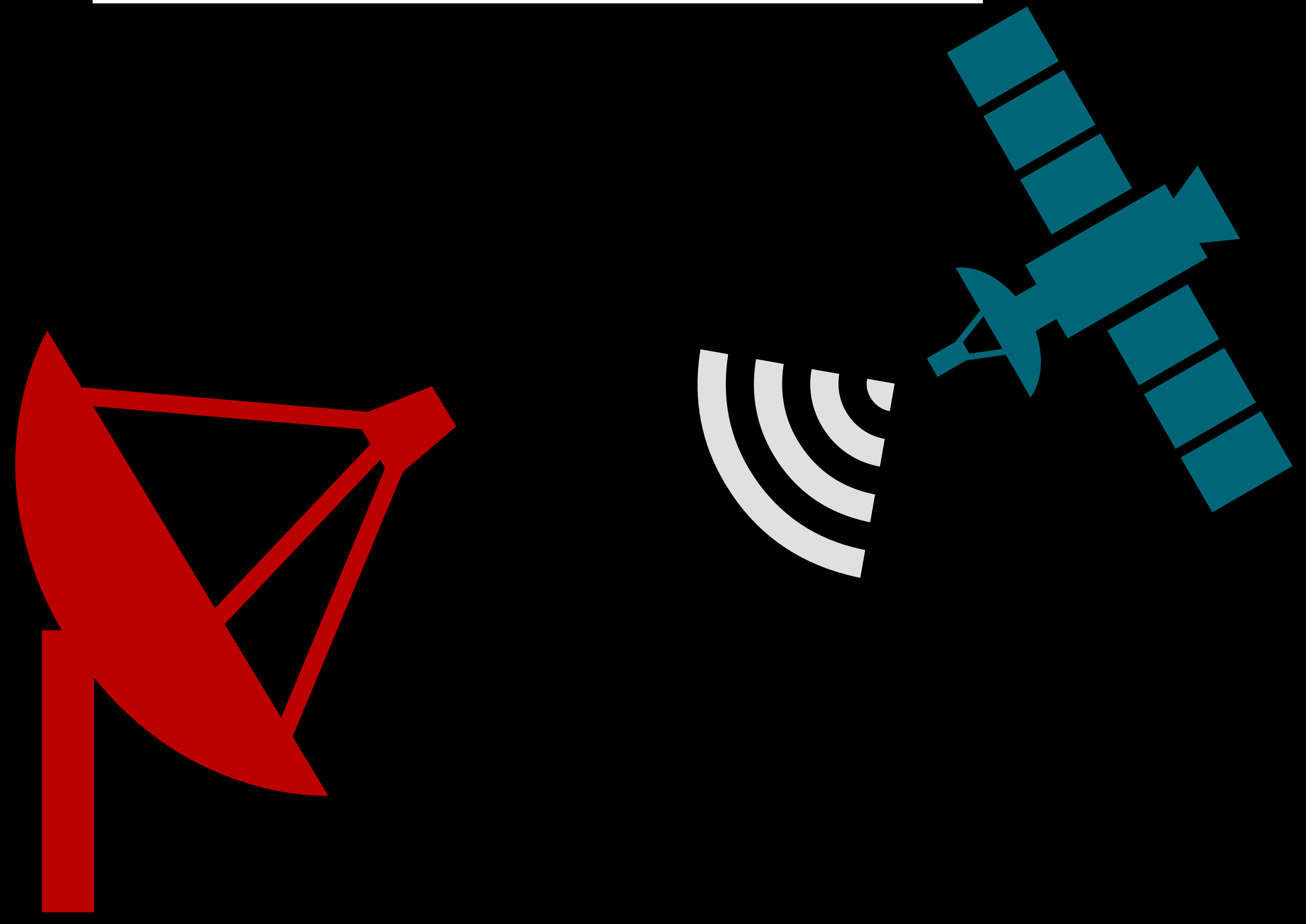
$$I_{\mathrm{D}} = \frac{P_{\mathrm{MODE}}}{v(t)}$$

Components of Space Simulation

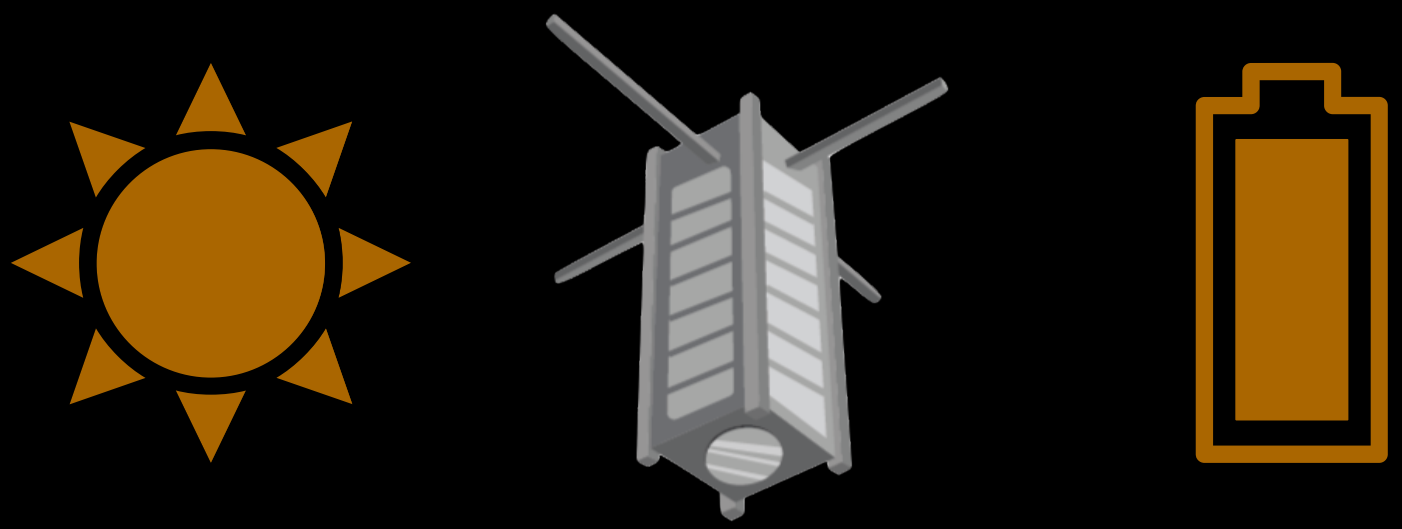
Orbital Mechanics



Communication



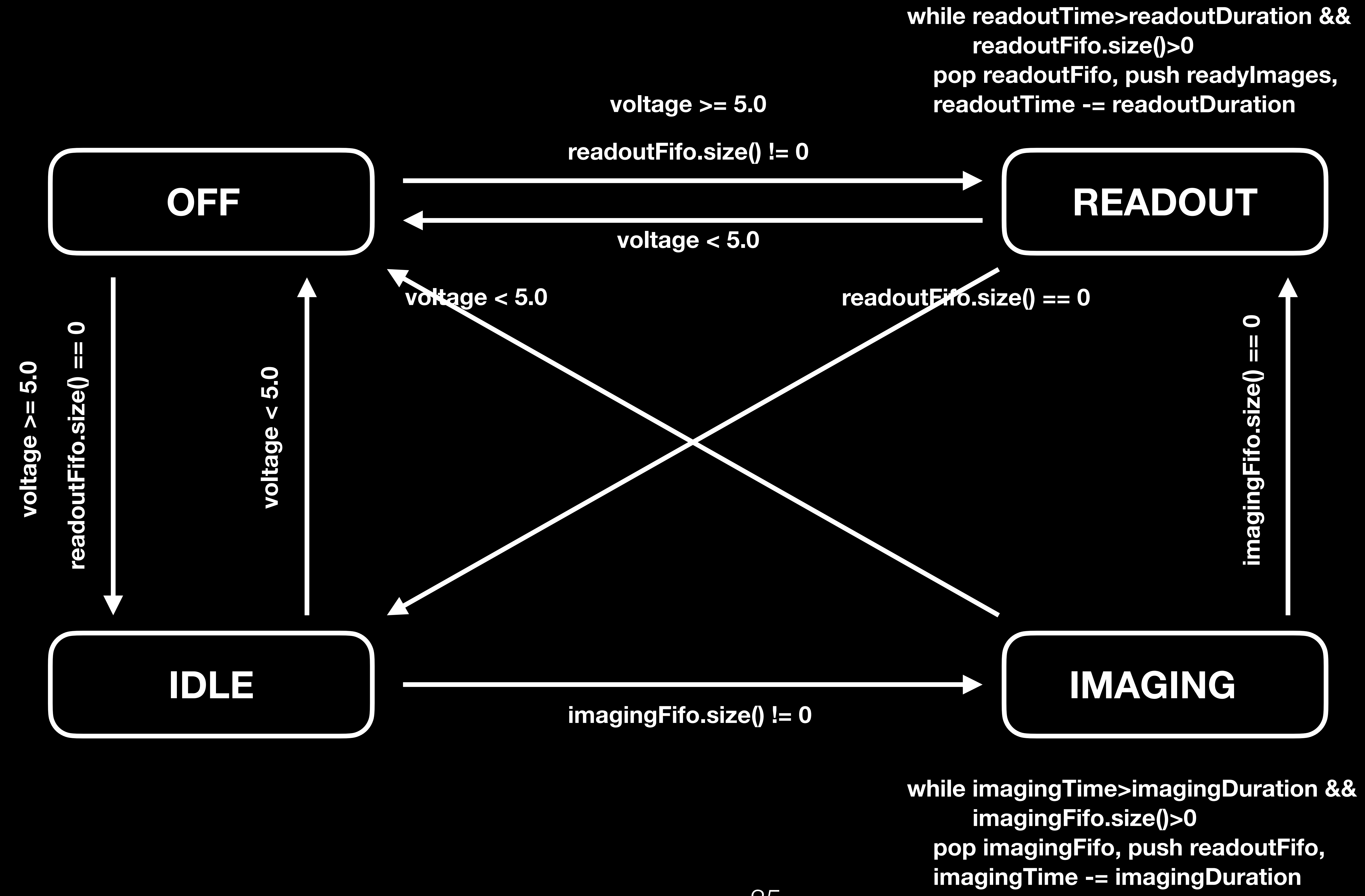
Energy and Power



Satellite Subsystems

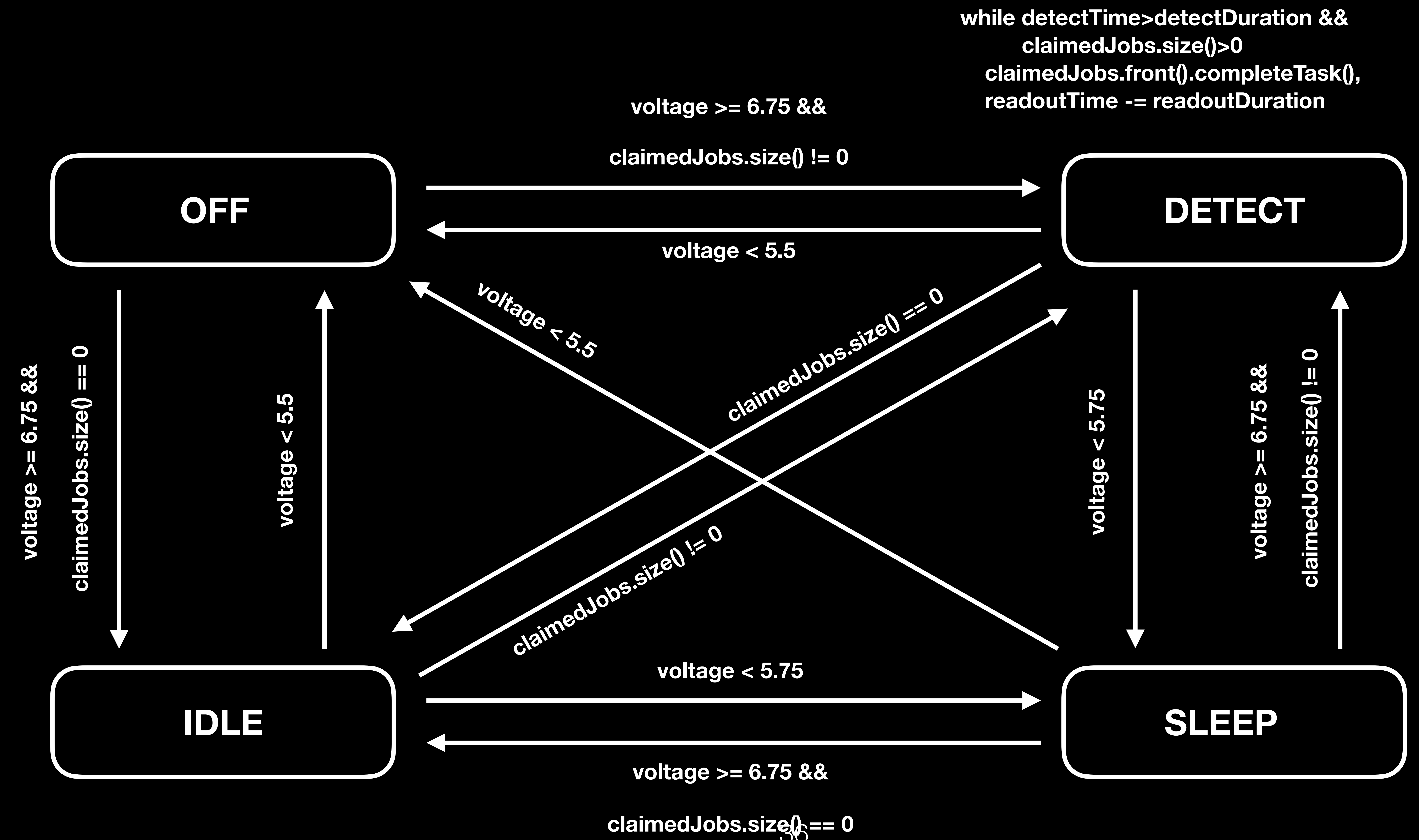
Camera Peripheral Modeling

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

Volume 11 / Number 10 / October, 1968

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

Volume 11 / Number 10 / October, 1968

Coordinate Frames

- ECI: Earth-centered inertial
  - (x,y,z) 3D coordinate frame
  - Usefule 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

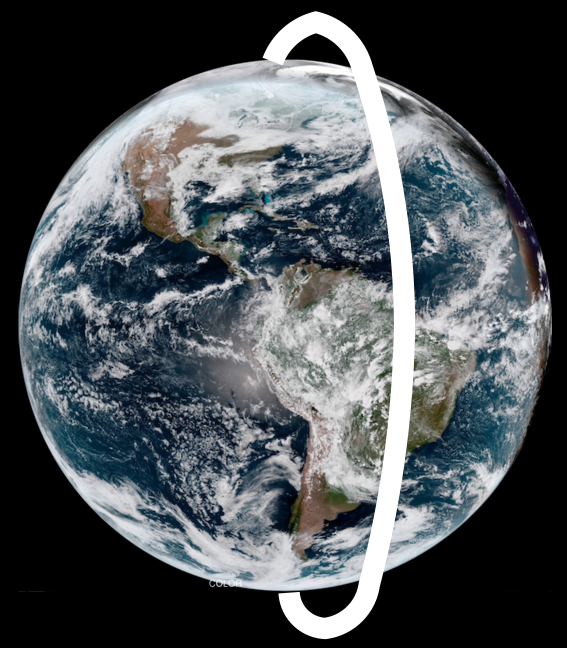
The `cote` Workflow

Example Programs in `cote`

The `cote` Workflow

The `cote` Workflow

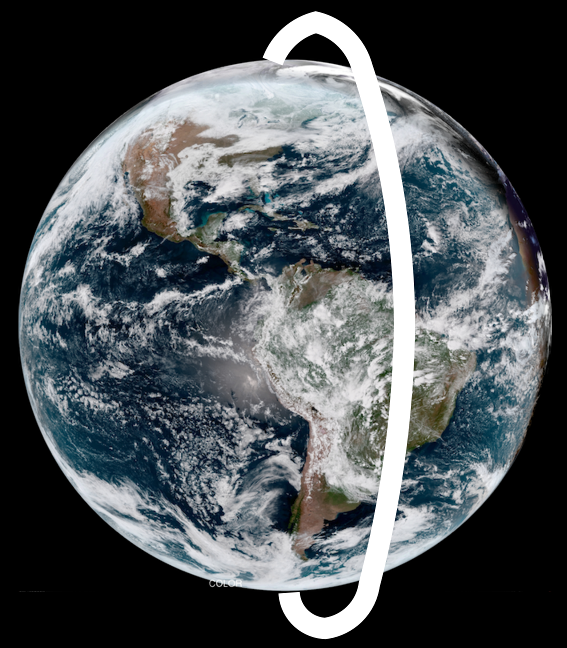
Simulator Inputs



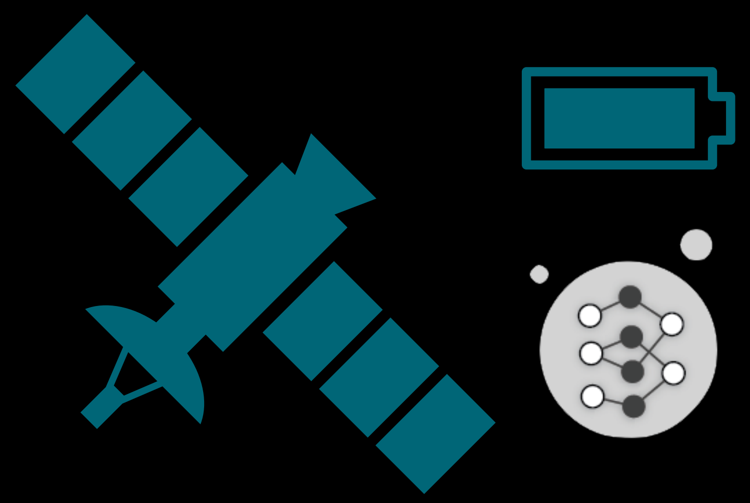
Orbital Mechanics Configuration

The `cote` Workflow

<u>Simulator Inputs</u>



Orbital Mechanics Configuration

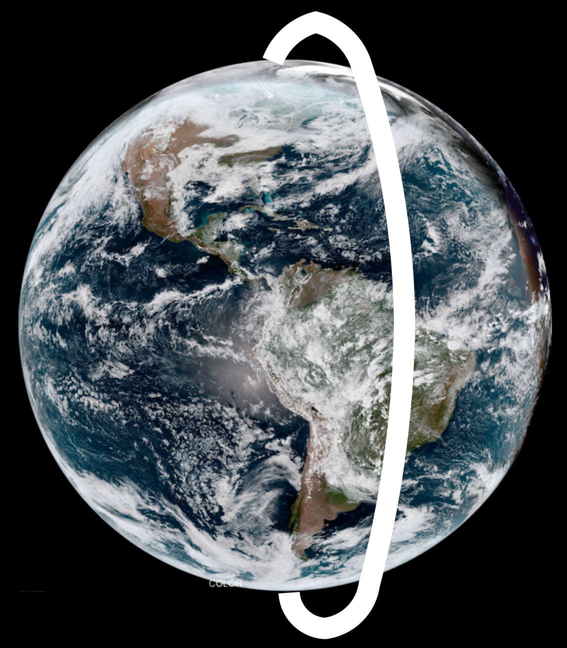


Satellite Configuration:

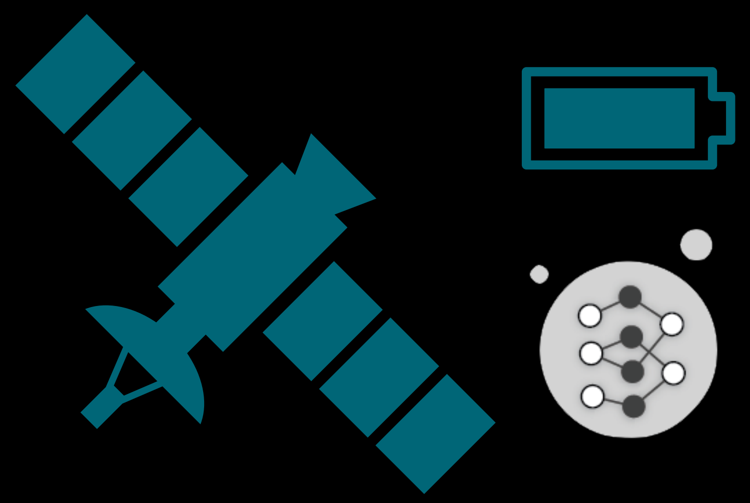
- **Energy System**
- **Onboard Processing**
- **Other Peripherals**

The `cote` Workflow

Simulator Inputs

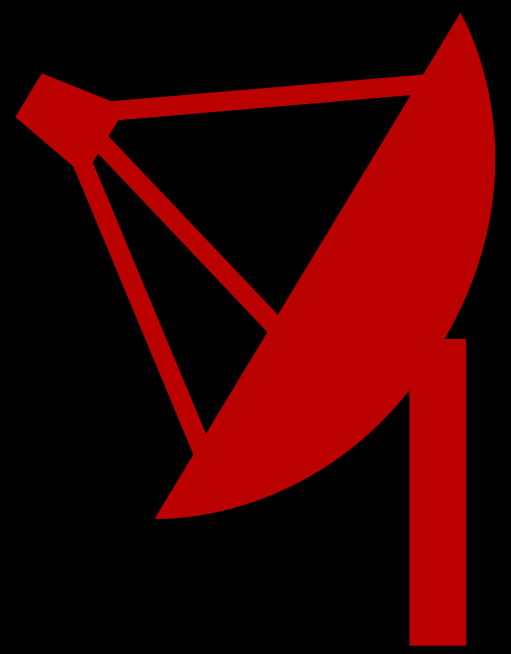


Orbital Mechanics Configuration



Satellite Configuration:

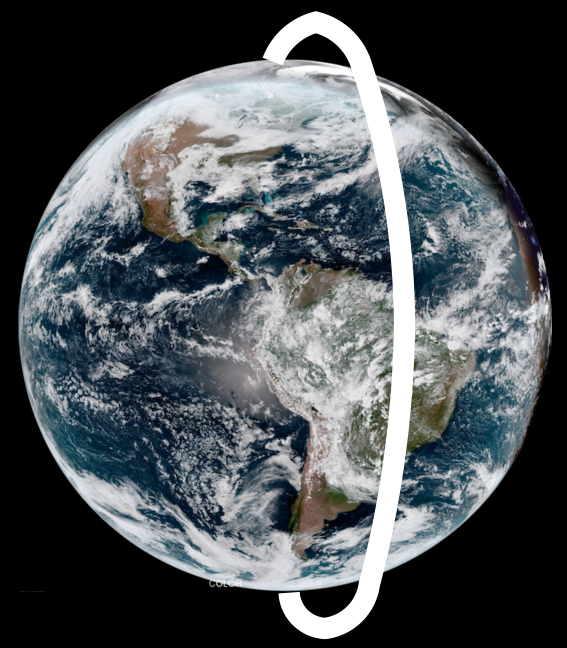
- **Energy System**
- **Onboard Processing**
- **Other Peripherals**



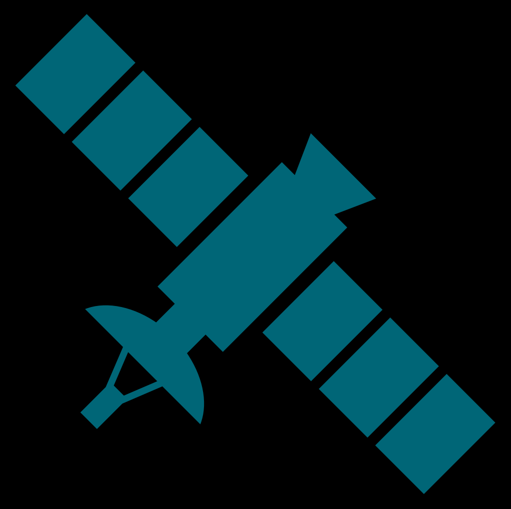
Communications Configuration

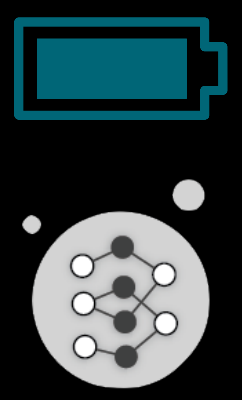
The `cote` Workflow

Simulator Inputs



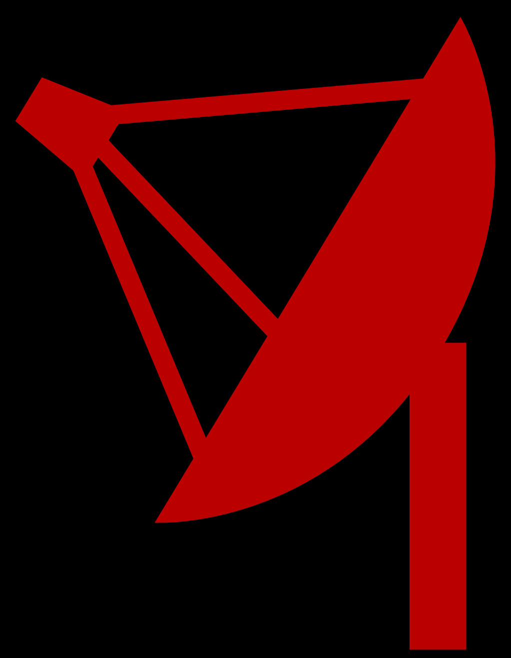
Orbital Mechanics Configuration





Satellite Configuration:

- **Energy System**
- **Onboard Processing**
- **Other Peripherals**



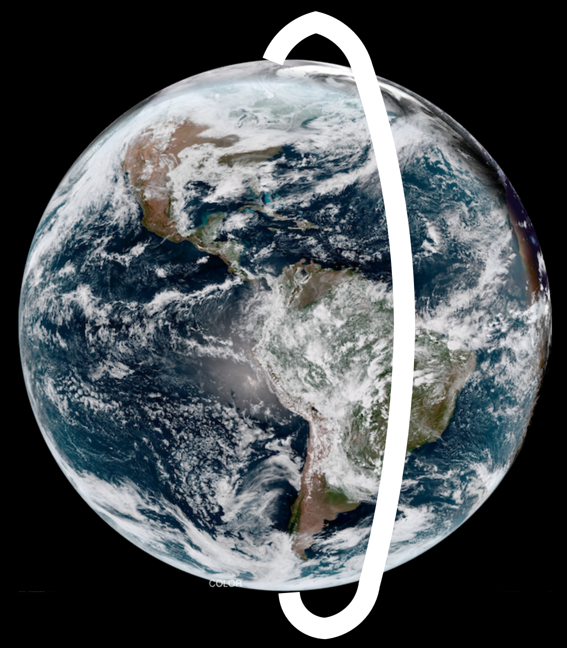
Communications Configuration

Simulator Outputs

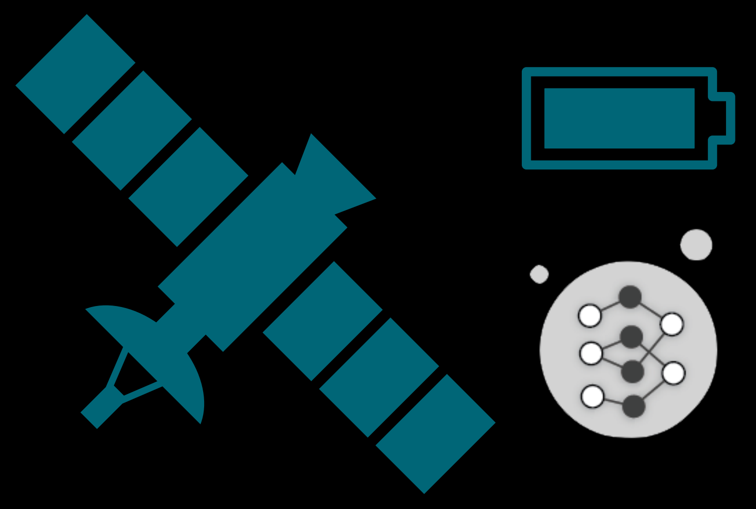
Scenario Simulation

The cote Workflow

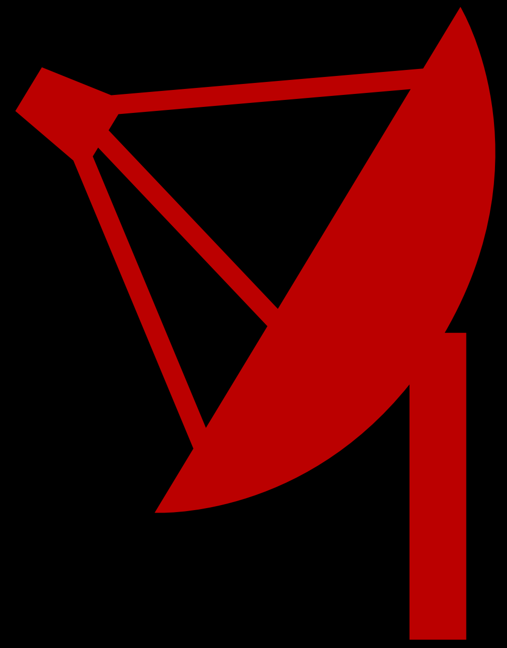
Simulator Inputs



Orbital Mechanics
Configuration



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

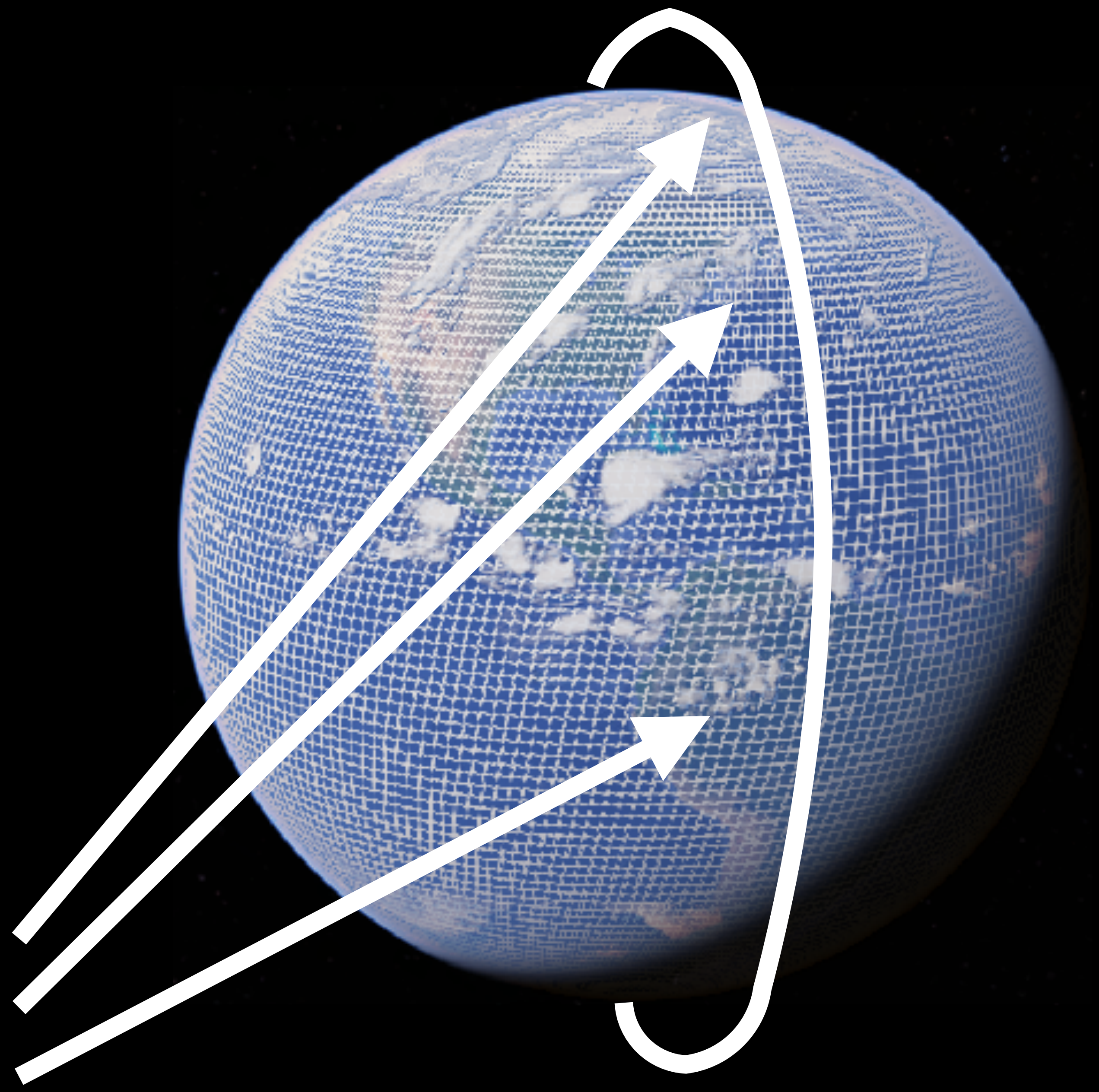


Communications
Configuration

Simulator Outputs

Scenario
Simulation

Event Traces



Observation Dates and Times
Satellite Activity Traces
Communication Logs

Example Satellite Config. File

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

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

Line: 1 Plain Text Soft Tabs: 2

Example Satellite Config. File

Satellite Epoch

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

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

Line: 1 Plain Text Soft Tabs: 2

Example Satellite Config. File

Satellite Epoch

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

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

Satellite TLE

Example Comms. Config. Files

Example Comms. Config. Files

tx-sat-0449380000.dat — configuration (git: master)

```
id,power,line_loss,gain,center_frequency,bandwidth
0449380000,+00002.000000000,-01.000000000,+06.000000000,008150000000.000000000,0000020000000.000000000
```

Line: 1 Plain Text Soft Tabs: 2

Example Comms. Config. Files

tx-sat-0449380000.dat — configuration (git: master)

```
id,power,line_loss,gain,center_frequency,bandwidth
0449380000,+00002.000000000,-01.000000000,+06.000000000,008150000000.000000000,0000020000000.000000000
```

Line: 1 Plain Text Soft Tabs: 2

rx-gnd-0000022115.dat — configuration (git: master)

```
id,gain,center-frequency,bandwidth
0000022115,+44.100000000,008150000000.000000000,0000020000000.000000000
```

Line: 1 Plain Text Soft Tabs: 2

Example Comms. Config. Files

tx-sat-0449380000.dat — configuration (git: master)

```
id,power,line_loss,gain,center_frequency,bandwidth
0449380000,+00002.000000000,-01.000000000,+06.000000000,008150000000.000000000,0000020000000.000000000
```

Line: 1 Plain Text Soft Tabs: 2

rx-gnd-0000022115.dat — configuration (git: master)

```
id,gain,center-frequency,bandwidth
0000022115,+44.100000000,008150000000.000000000,0000020000000.000000000
```

Line: 1 Plain Text Soft Tabs: 2

gnd-0000022115-the-university-of-hong-kong.gnd — configuration (git: master)

```
id,lat,lon,hae-km
0000022115,+22.283888889,+114.138055556,+00.000000000
```

Line: 1 Plain Text Soft Tabs: 2

Example Energy Config. Files

Example Energy Config. Files

solar-array-sat-0449380000.dat — configuration (git: master)

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

Line: 1 Plain Text Soft Tabs: 2

Example Energy Config. Files

solar-array-sat-0449380000.dat — configuration (git: master)

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

Line: 1 | Plain Text | Soft Tabs: 2

capacitor-sat-0449380000.dat — configuration (git: master)

```
capacitance-f,esr-ohm,charge-c,id
000005.000000000000,000.168000,000000034.3021440000000000000000,0449380000
```

Line: 1 | Plain Text | Soft Tabs: 2

Example Subsystem Config. Files

Three-state ADACS State Machine

adacs-sm-sat-0449380000.dat — configuration (git: master)

```
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
```

Line: 1 Plain Text Soft Tabs: 2

Example Subsystem Config. Files

Imager peripheral:

```
constant:imaging-duration-s:0.031260
constant:readout-duration-s:0.838860
variable:imaging-time-s:0.0
variable:readout-time-s:0.0
variable:power-w:0.0
variable:node-voltage:0.0
variable:imaging-task-count:0
variable:readout-task-count:0
variable:ready-images:0
enter-state:OFF:power-w=0.0
enter-state:IDLE:power-w=0.0
enter-state:IMAGING:power-w=3.5
enter-state:READOUT:power-w=2.5
exit-state:IMAGING:imaging-time-s=0.0
transition:OFF->READOUT:node-voltage>=5.0,readout-task-count!=0
transition:OFF->IDLE:node-voltage>=5.0,readout-task-count==0
transition:IDLE->OFF:node-voltage<5.0
transition:IDLE->IMAGING:node-voltage>=5.0,imaging-task-count!=0
transition:IMAGING->OFF:node-voltage<5.0:imaging-task-count=0
transition:IMAGING->READOUT:node-voltage>=5.0,imaging-task-count==0
transition:READOUT->OFF:node-voltage<5.0
transition:READOUT->IDLE:node-voltage>=5.0,readout-task-count==0:readout-time-s=0.0
initial-state:OFF
```

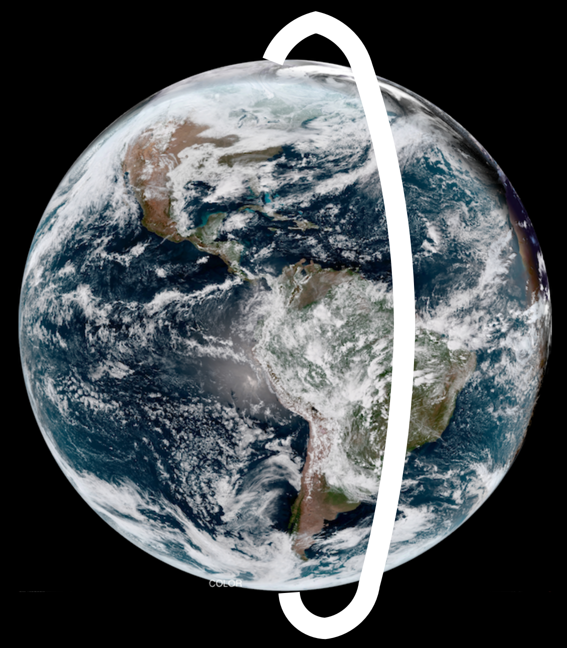
Example Subsystem Config. Files

Processor peripheral running object detection workload:

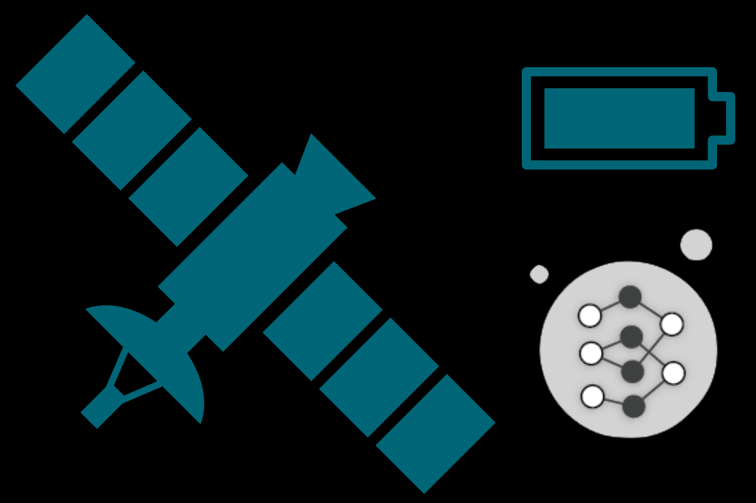
```
constant:task-duration-s:0.044860
variable:work-time-s:0.0
variable:power-w:0.0
variable:node-voltage:0.0
variable:claimed-task-count:0
variable:transmitting:0
enter-state:OFF:power-w=0.0
enter-state:SLEEP:power-w=0.5
enter-state:IDLE:power-w=0.5
enter-state:PAUSE:power-w=0.5
enter-state:WORK:power-w=11.3272
exit-state:WORK:work-time-s=0.0
transition:OFF->IDLE:node-voltage>=6.75,claimed-task-count==0
transition:OFF->PAUSE:node-voltage>=6.75,claimed-task-count!=0,transmitting!=0
transition:OFF->WORK:node-voltage>=6.75,claimed-task-count!=0,transmitting==0
transition:SLEEP->OFF:node-voltage<5.5
transition:SLEEP->IDLE:node-voltage>=6.75,claimed-task-count==0
transition:SLEEP->PAUSE:node-voltage>=6.75,claimed-task-count!=0,transmitting!=0
transition:SLEEP->WORK:node-voltage>=6.75,claimed-task-count!=0,transmitting==0
transition:IDLE->OFF:node-voltage<5.5
transition:IDLE->SLEEP:node-voltage<5.75,node-voltage>=5.5
transition:IDLE->PAUSE:node-voltage>=5.75,claimed-task-count!=0,transmitting!=0
transition:IDLE->WORK:node-voltage>=5.75,claimed-task-count!=0,transmitting==0
transition:PAUSE->OFF:node-voltage<5.5
transition:PAUSE->SLEEP:node-voltage<5.75,node-voltage>=5.5
transition:PAUSE->IDLE:node-voltage>=5.75,claimed-task-count==0
transition:PAUSE->WORK:node-voltage>=5.75,claimed-task-count!=0,transmitting==0
transition:WORK->OFF:node-voltage<5.5
transition:WORK->SLEEP:node-voltage<5.75,node-voltage>=5.5
transition:WORK->IDLE:node-voltage>=5.75,claimed-task-count==0
transition:WORK->PAUSE:node-voltage>=5.75,claimed-task-count!=0,transmitting!=0
initial-state:OFF
```

The `cote` Workflow

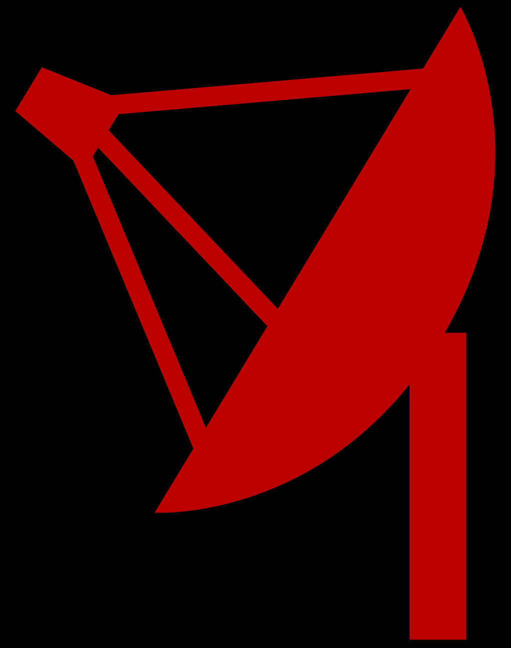
Simulator Inputs



Orbital Mechanics Configuration



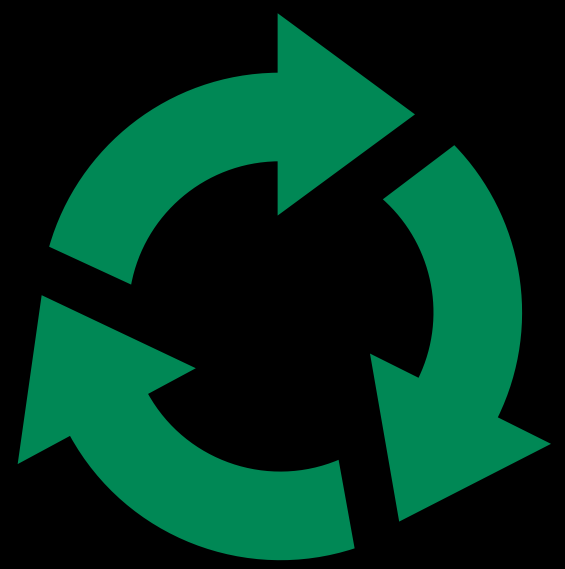
Satellite Configuration:
- Energy System
- Onboard Processing
- Other Peripherals



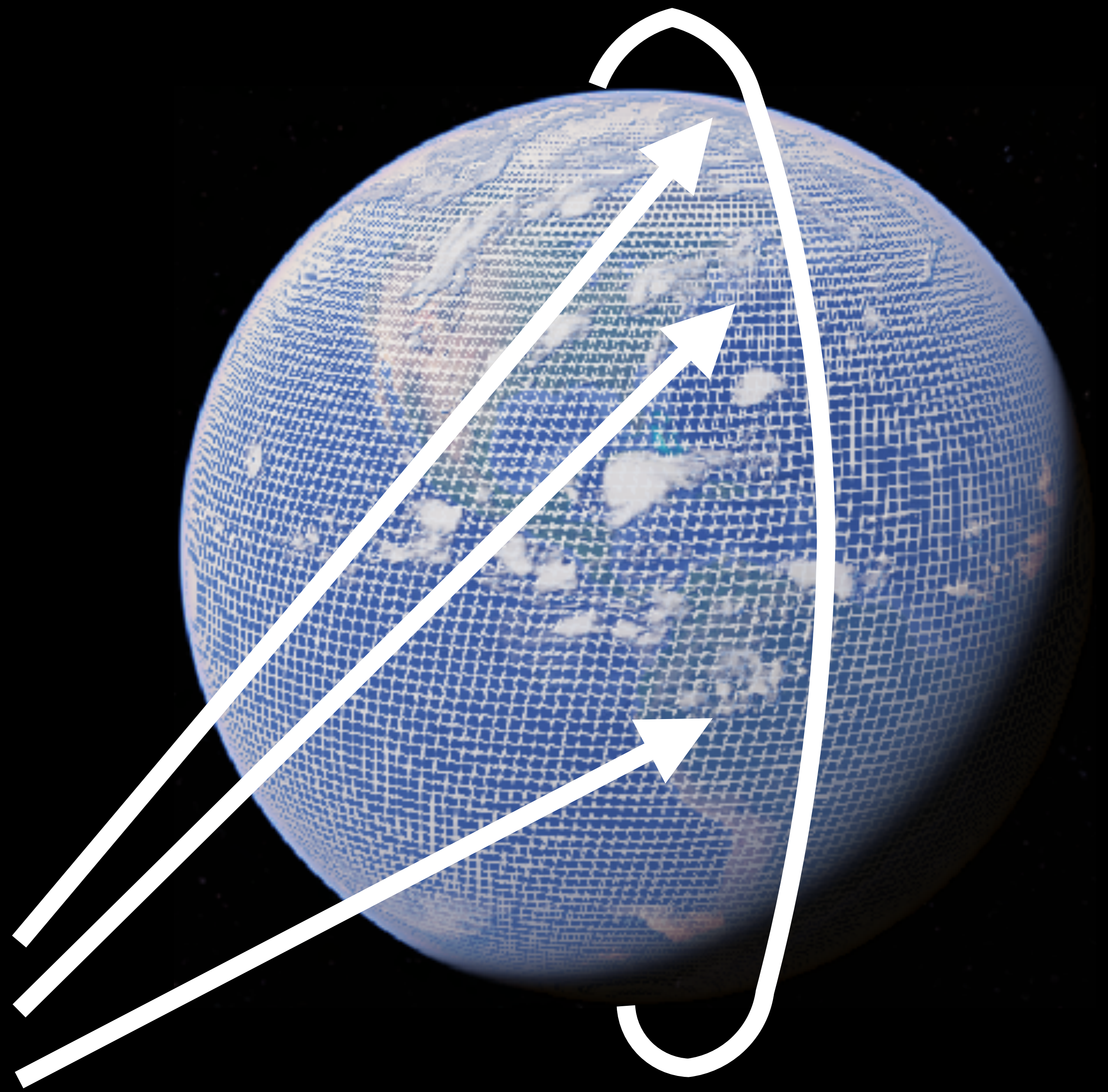
Communications Configuration

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

meas-sat-eci.csv — materials

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

Line: 1 CSV Soft Tabs: 2

Overview

Background and Related Work

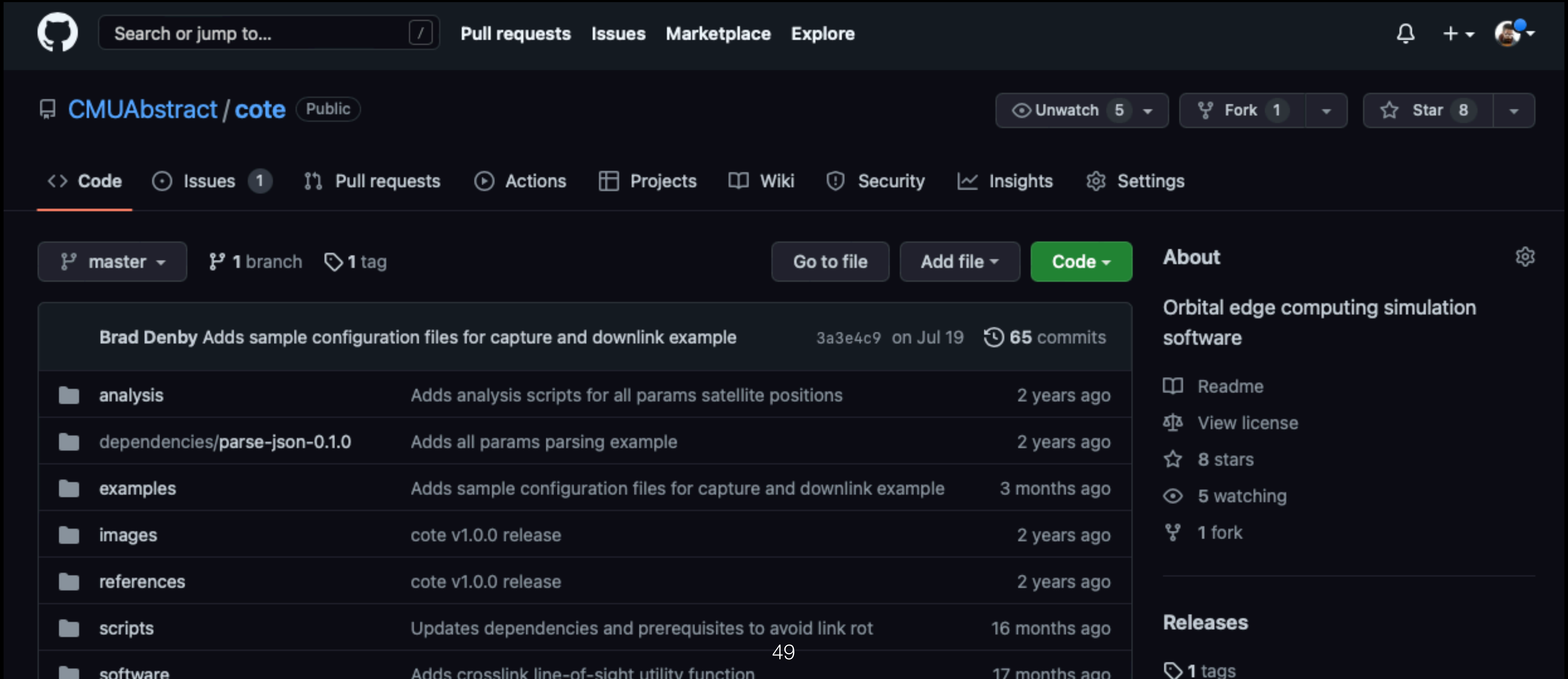
Components of Space Simulation

The `cote` Workflow

Example Programs in `cote`

Open Source Development

https://github.com/CMUAbstract/cote



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-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

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-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

Example Program: “Generate ECIs”

Input Configuration Files

Output Log Files

Example Program: “Generate ECIs”

Input Configuration Files

Output Log Files

Time Step: 1 minute

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

- ..
- 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-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

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-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

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-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

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-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

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-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

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

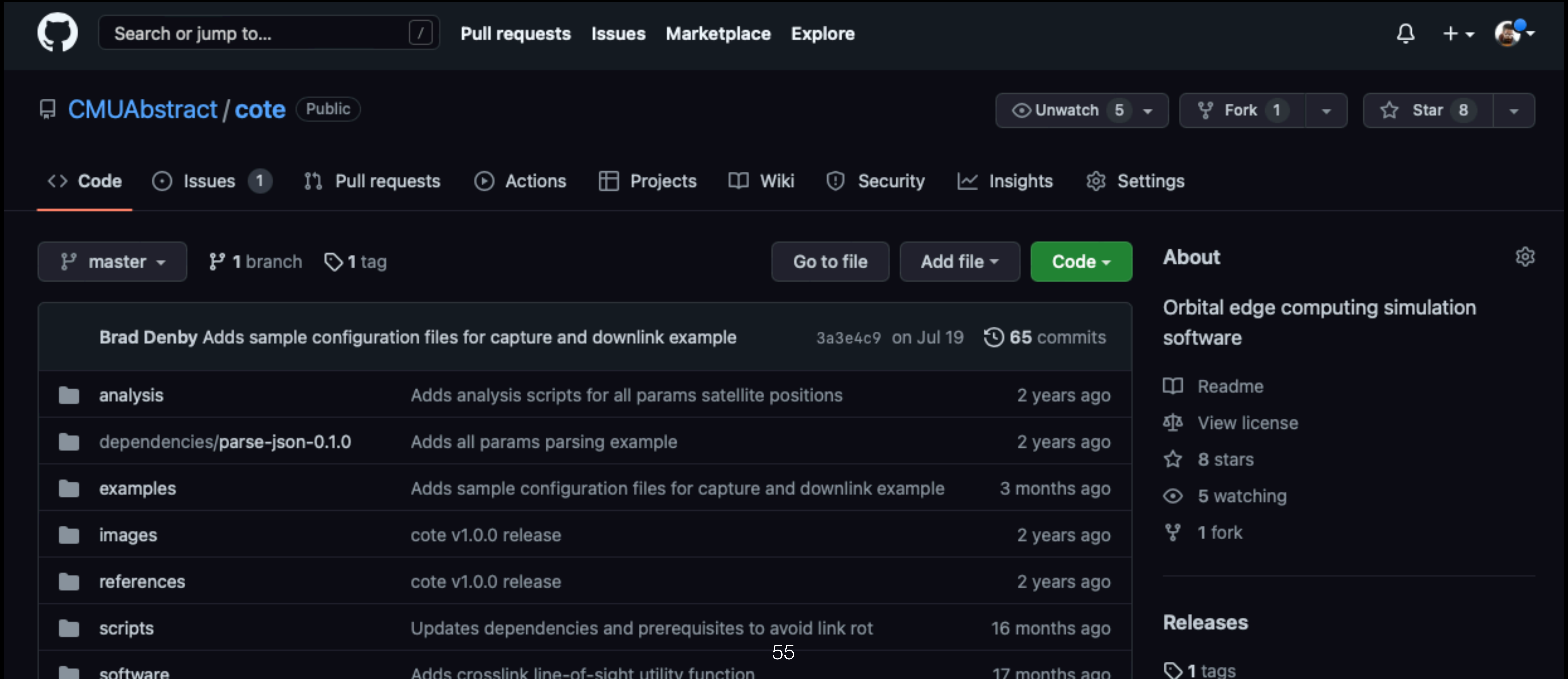
- ..
- all-params
- bent-pipe
- capture-and-downlink
- close-spaced
- crosslink-los
- dynamic-tiling
- frame-spaced
- generate-coms
- generate-constellation-configuration
- generate-constellation-tracks
- generate-deadlines
- generate-ecis
- generate-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

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-gnds
- generate-ground-tracks
- generate-irradiance-trace
- generate-links
- generate-sat-azel
- generate-sun-azel
- generate-voltage-trace
- README.md

Open Source Development

https://github.com/CMUAbstract/cote



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