

SpaceCoMP

Distributed On-Board Processing for Live Earth Observation

Klas Segeljakt^{1,2}, Arlette Houndji^{1,3}, Burcu Koçoğlu¹, Tobias Edman¹, Thomas Sandholm¹, Bernardo Huberman¹, Paris Carbone^{1,2}

¹RISE Research Institutes of Sweden, ²KTH Royal Institute of Technology, ³Karlstad University



Rymdstyrelsen
Swedish National Space Agency



Earth Observation (EO): From Curiosity to Necessity

Earth Observation (EO): From Curiosity to Necessity

Deforestation monitoring



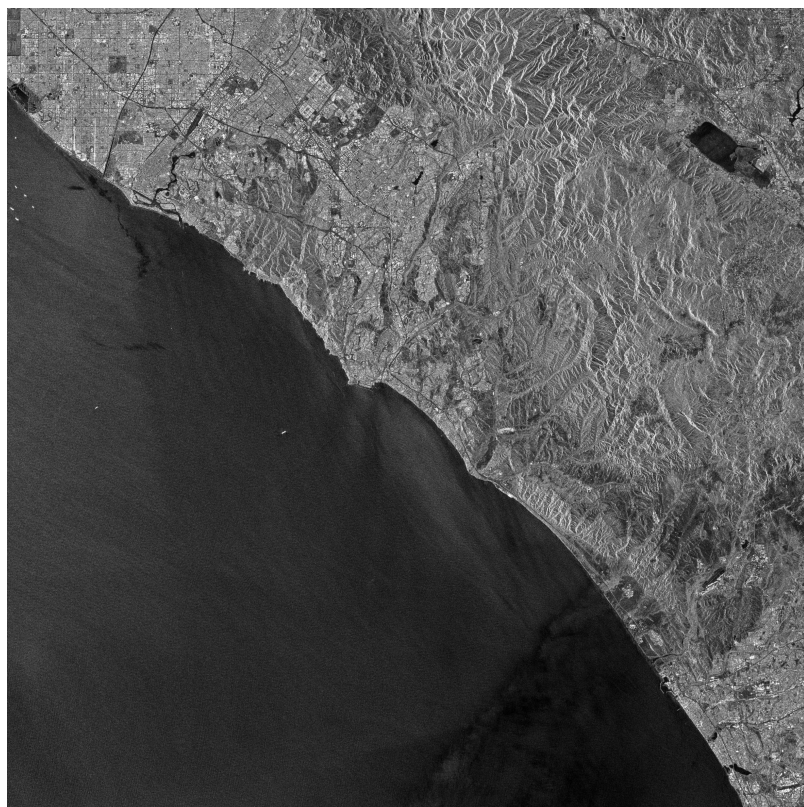
Rondônia, Brazil 2008
Source: NASA / Landsat 9

Crop monitoring



Finney County, Kansas USA 2001
Source: NASA / ASTER

Oil spill detection



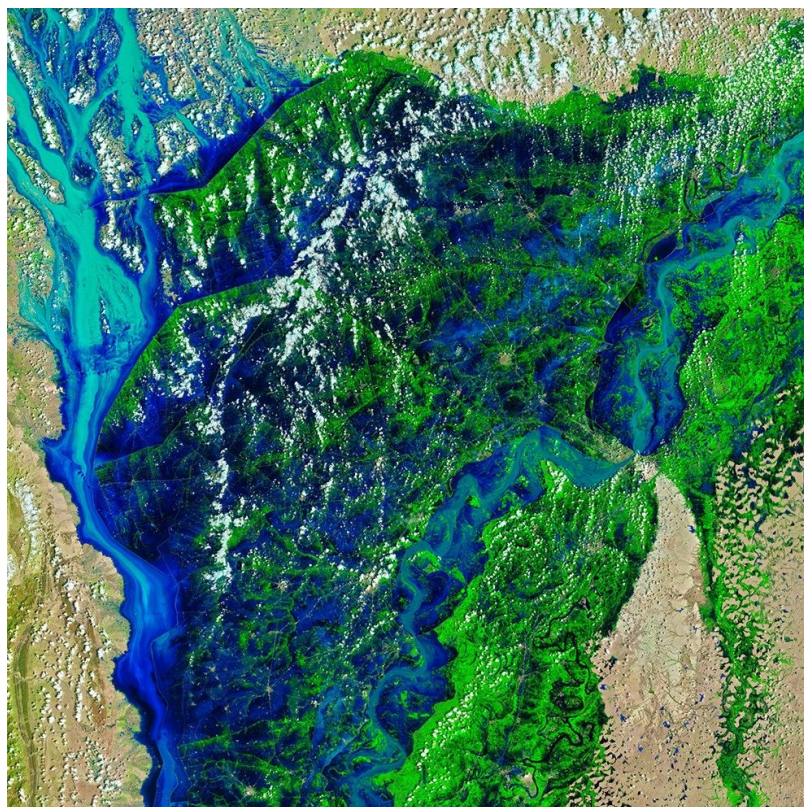
Huntington Beach, California USA 2021
Source: NASA / ESA Copernicus

Maritime tracking



Suez Canal, Egypt 2021
Source: NASA / Landsat 8

Flood response

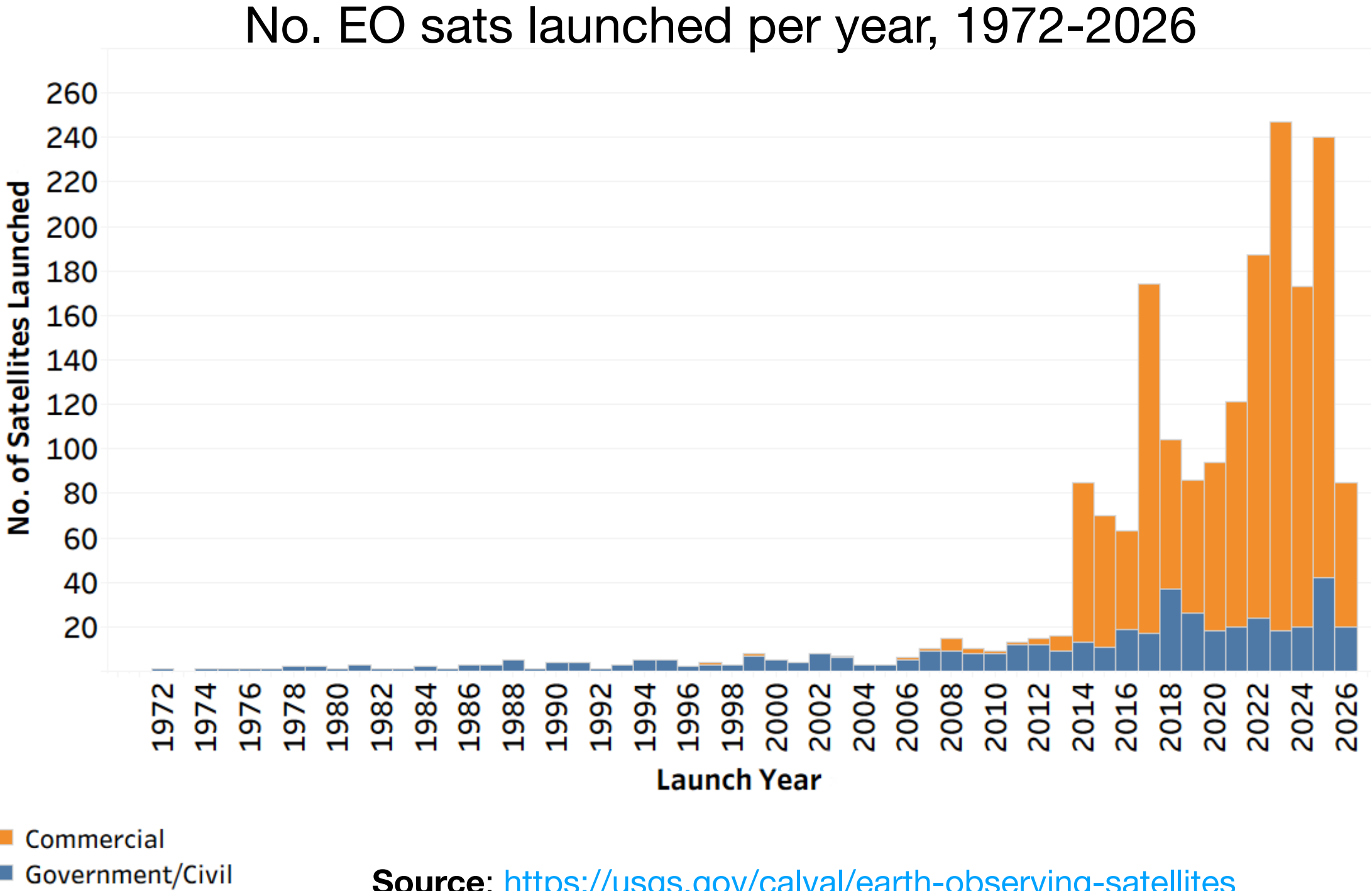


Indus river, Pakistan 2022
Source: NASA / Landsat 9

What Has Happened

What Has Happened

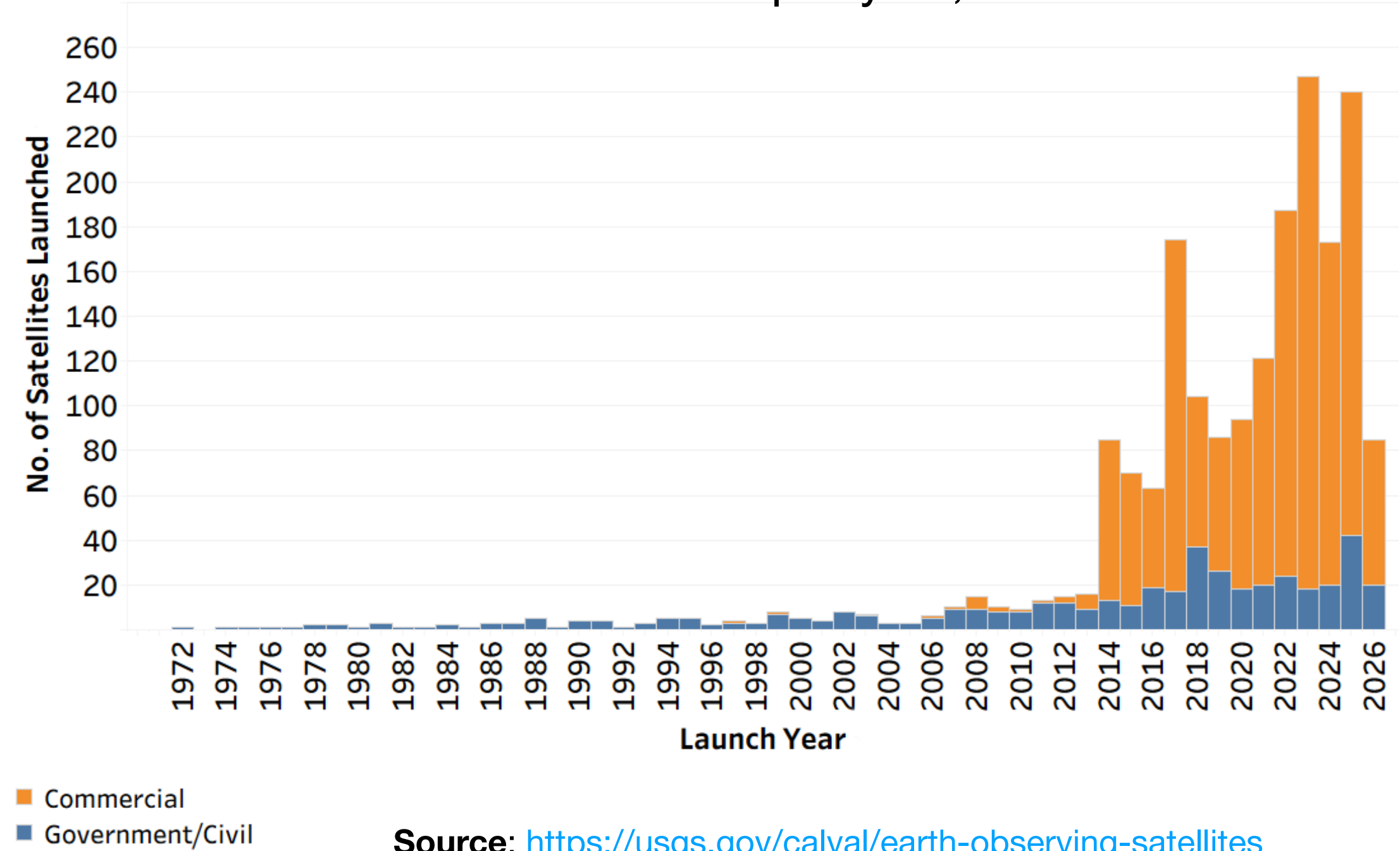
More Satellite Launches



What Has Happened

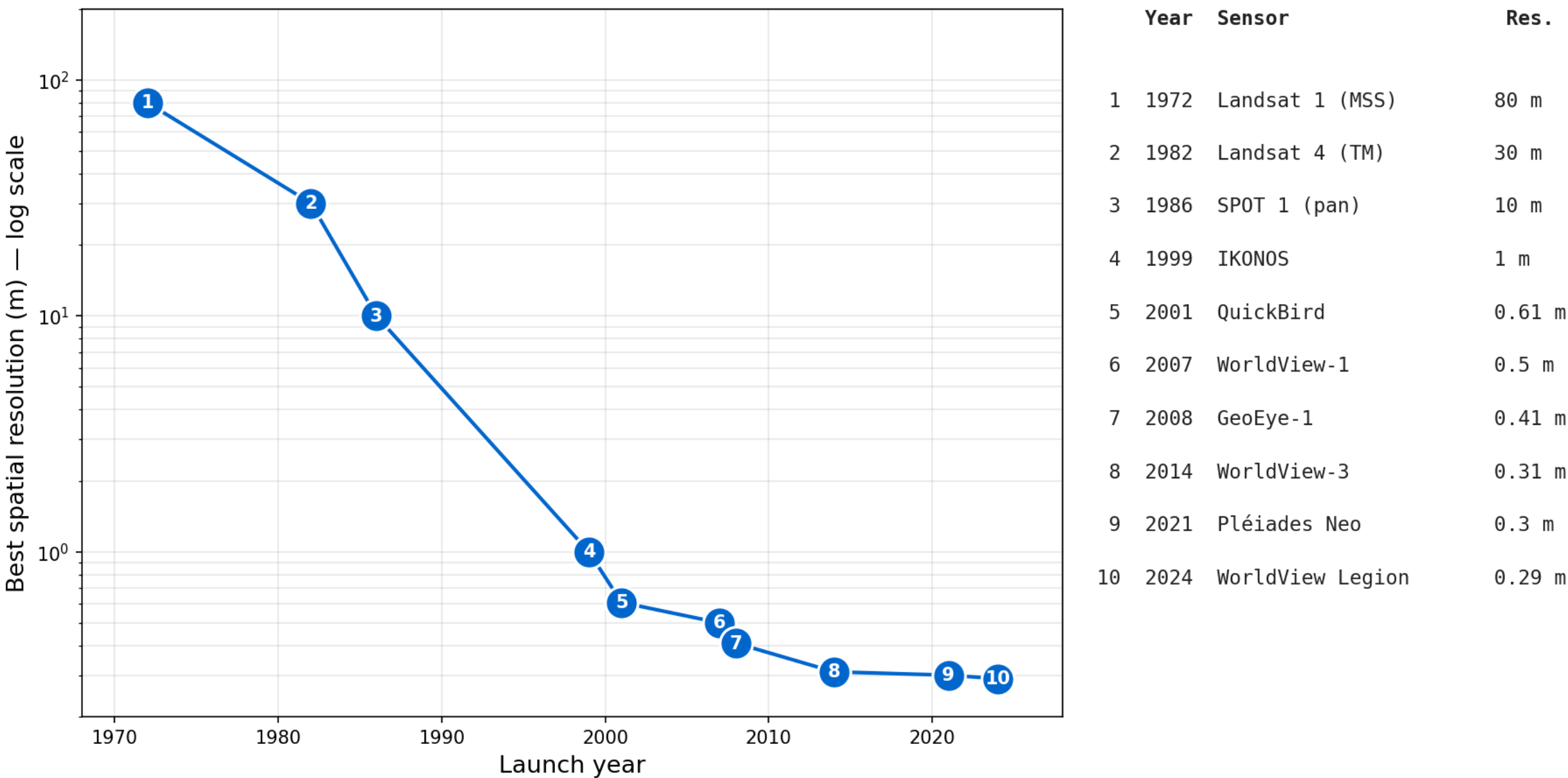
More Satellite Launches

No. EO sats launched per year, 1972-2026



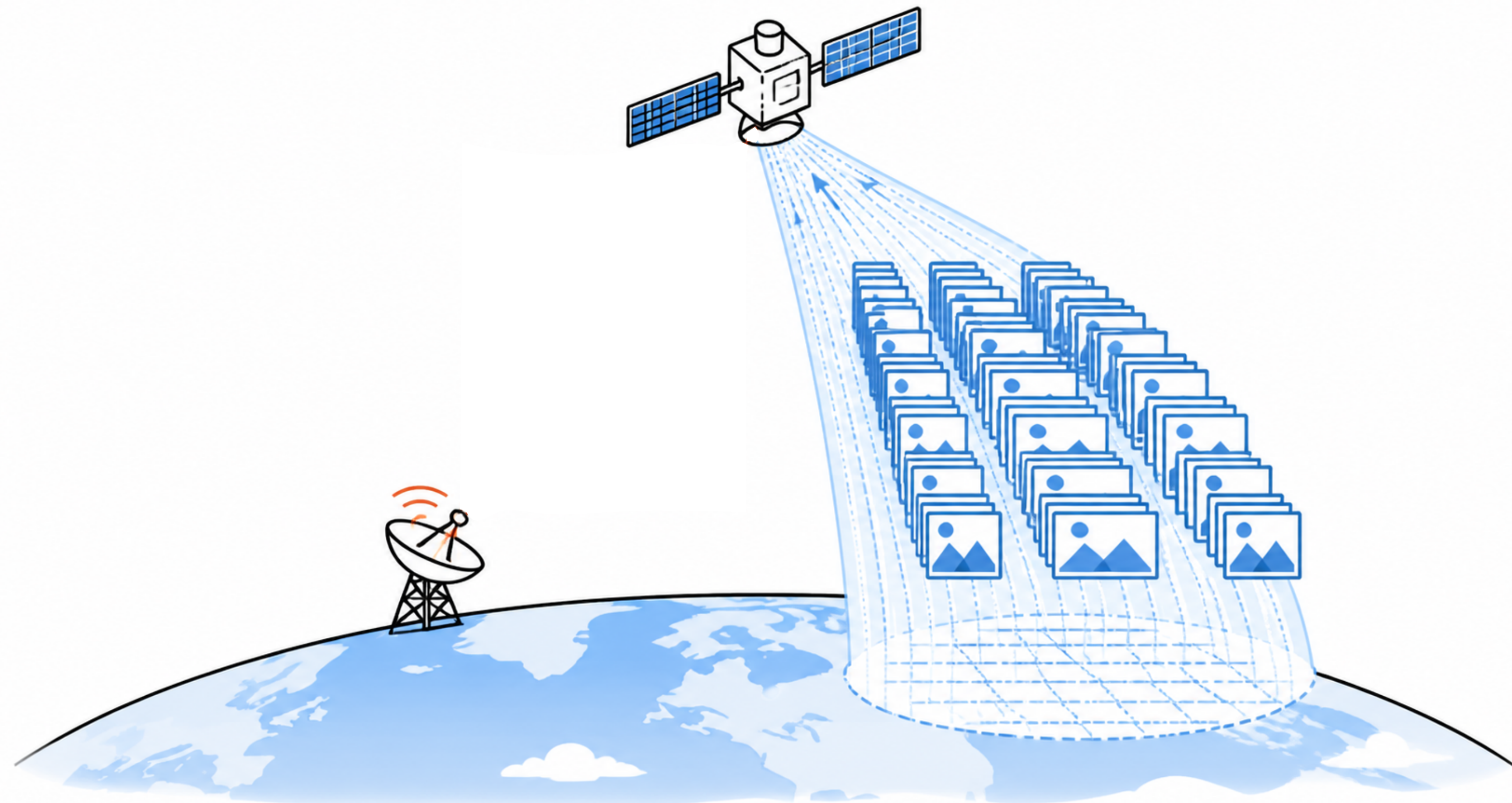
Improved Sensors

Optical Image Resolution of EO sats, 1972-2024



The Problem

A single LEO satellite can collect gigabytes to terabytes of data every day.

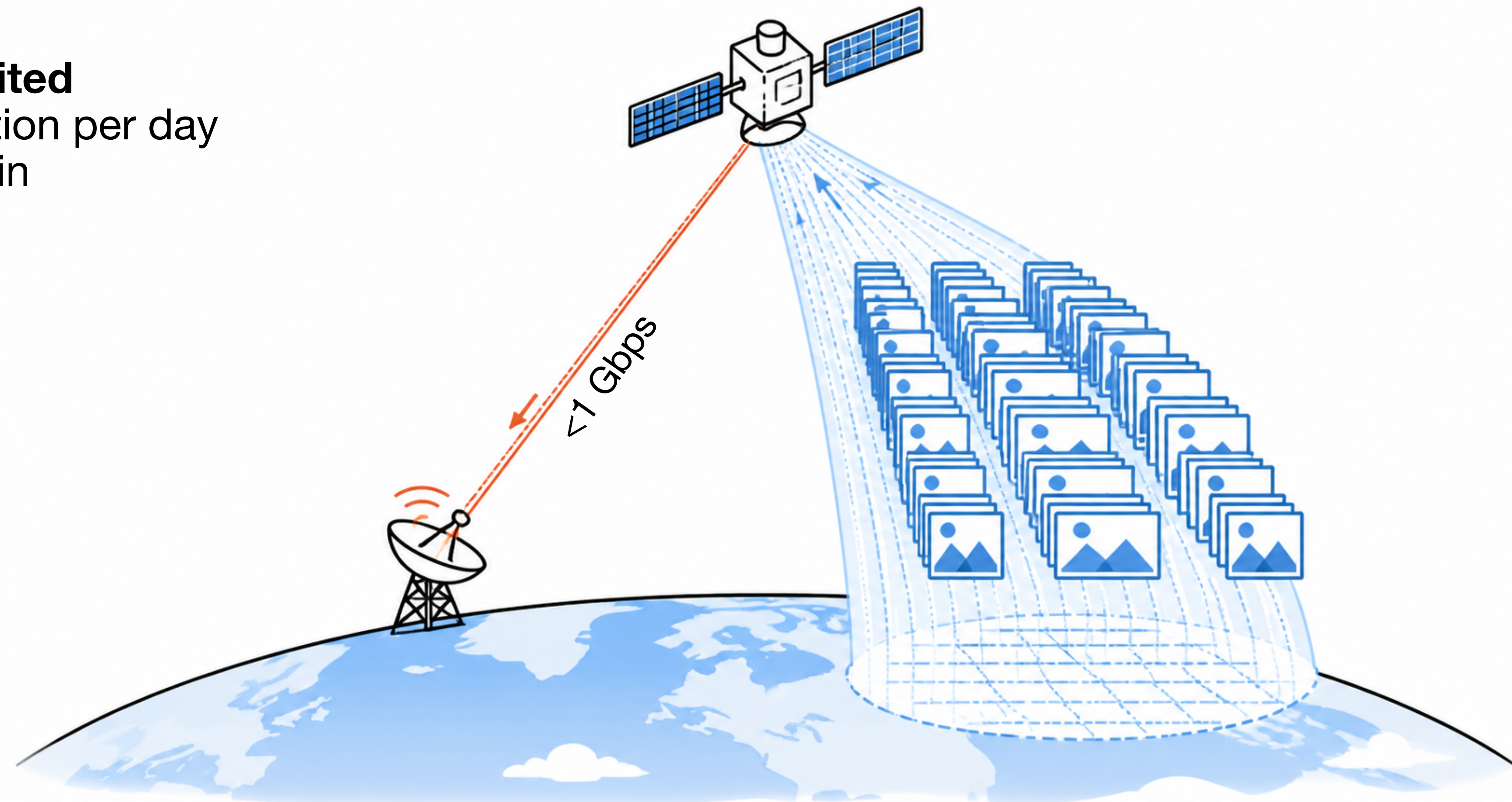


The Problem

A single LEO satellite can collect gigabytes to terabytes of data every day.

But, contact to ground is limited

- ➔ Only a few contacts per station per day
- ➔ Each contact lasts ~5–15 min



The Problem

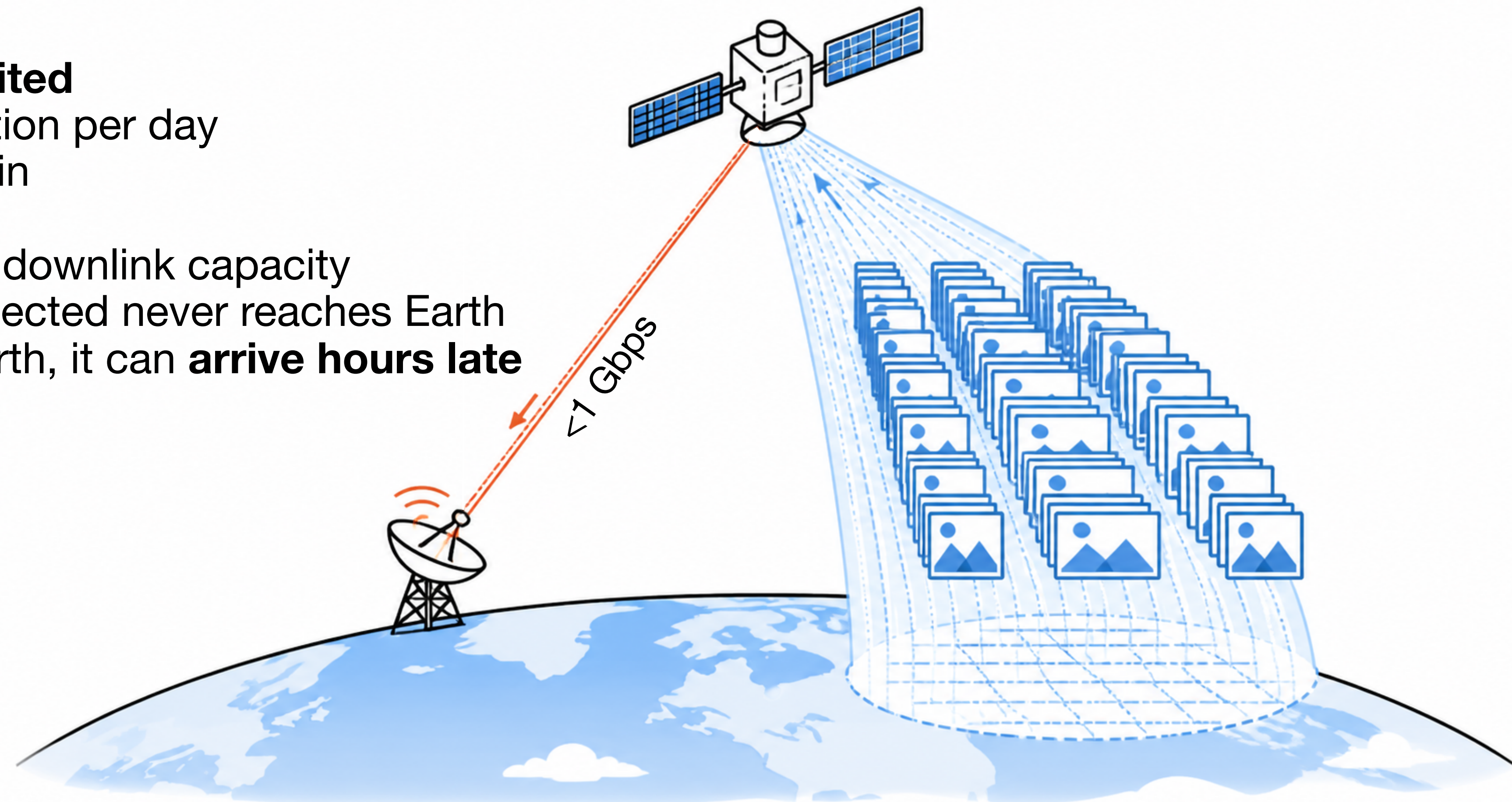
A single LEO satellite can collect gigabytes to terabytes of data every day.

But, contact to ground is limited

- ➡ Only a few contacts per station per day
- ➡ Each contact lasts ~5–15 min

Data collection is **throttled** by downlink capacity

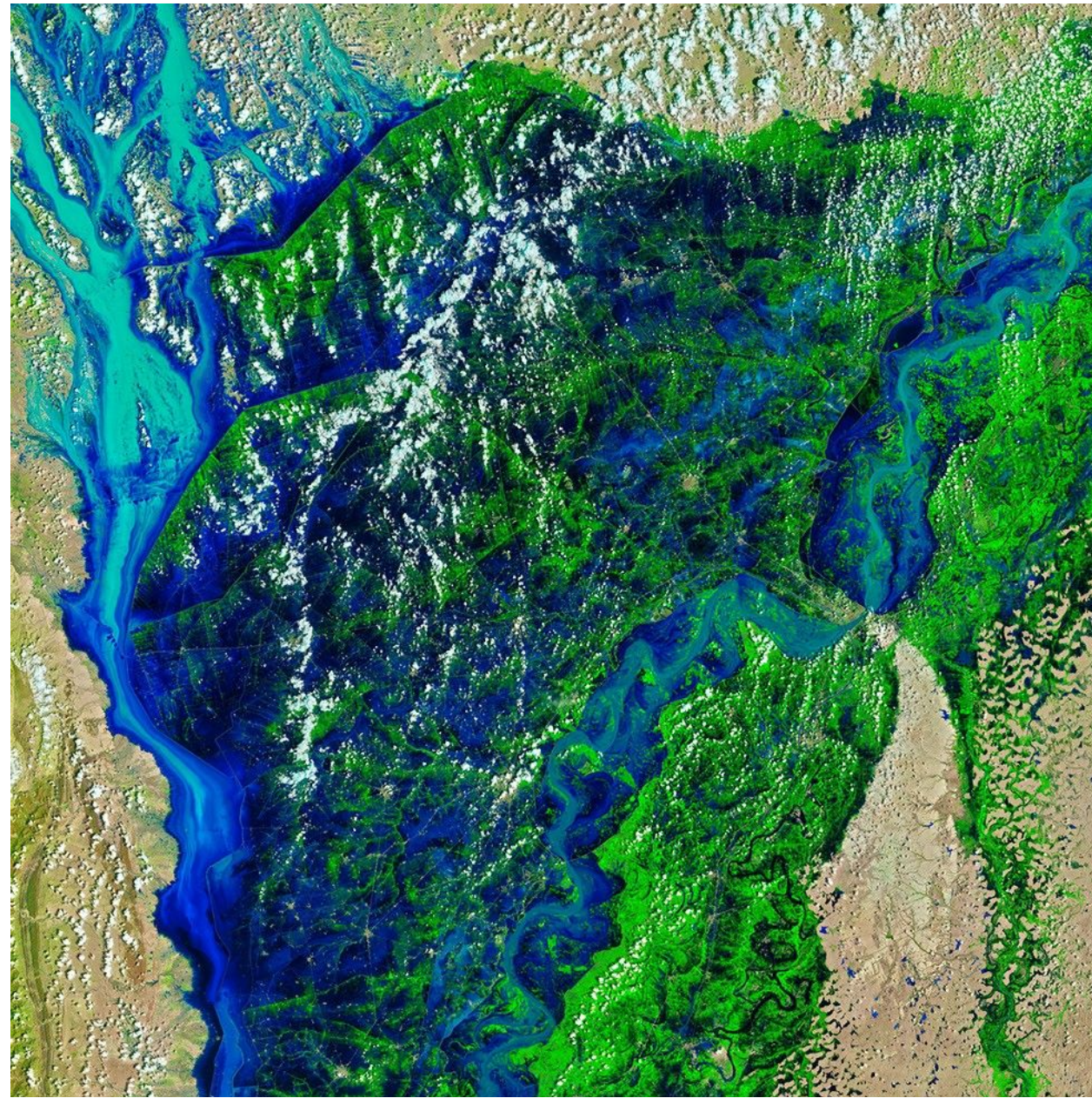
- ➡ Most data that *could be* collected never reaches Earth
- ➡ Even when data reaches Earth, it can **arrive hours late**



Why Time Matters

Why Time Matters

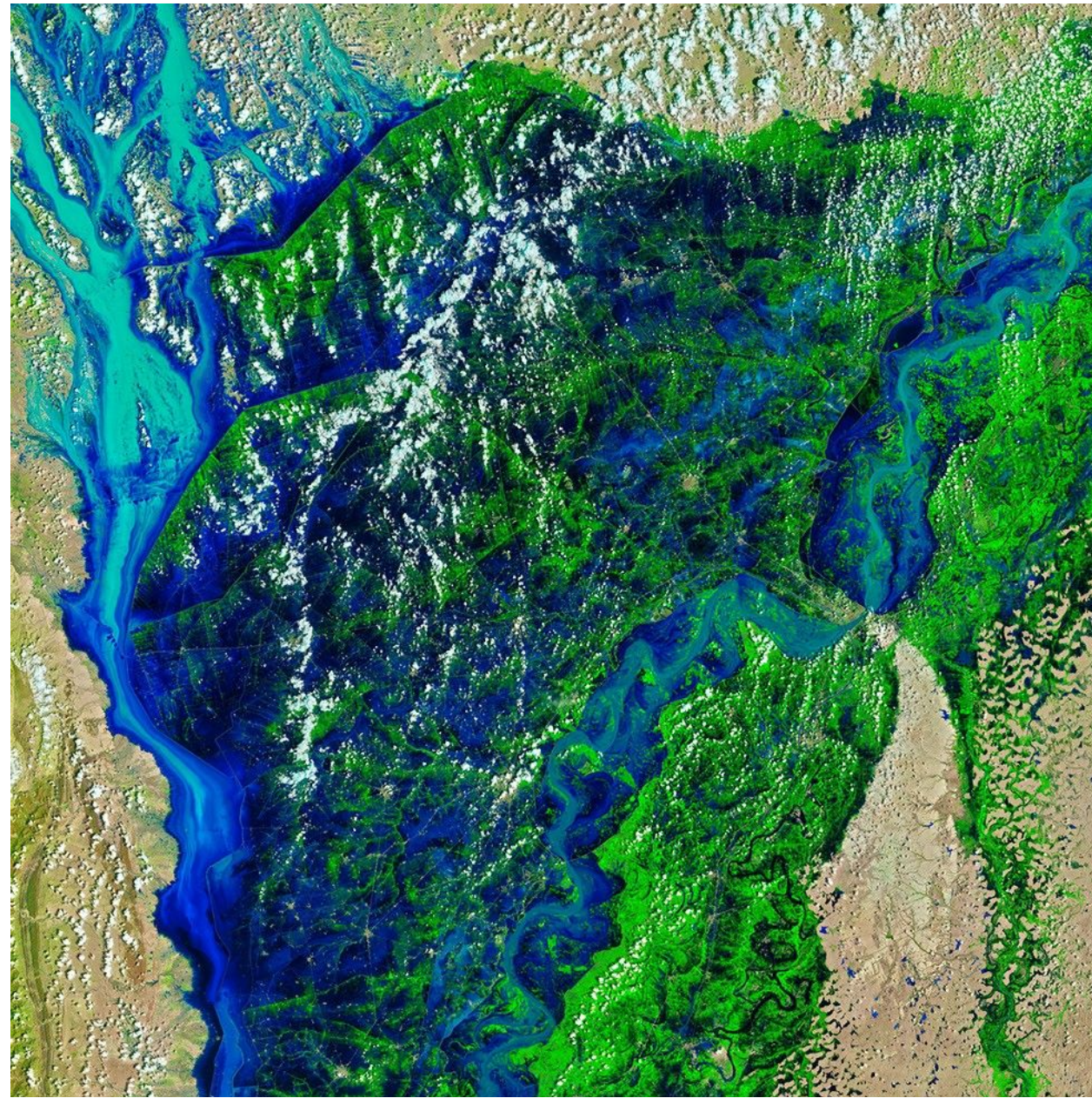
Flood response



Indus river, Pakistan 2022
Source: NASA / Landsat 9

Why Time Matters

Flood response



Indus river, Pakistan 2022
Source: NASA / Landsat 9

Weeks to Months

→ What needs rebuilding?

Hours to Days

→ Which roads are still passable?

Seconds to Minutes

→ Who needs warning right now?

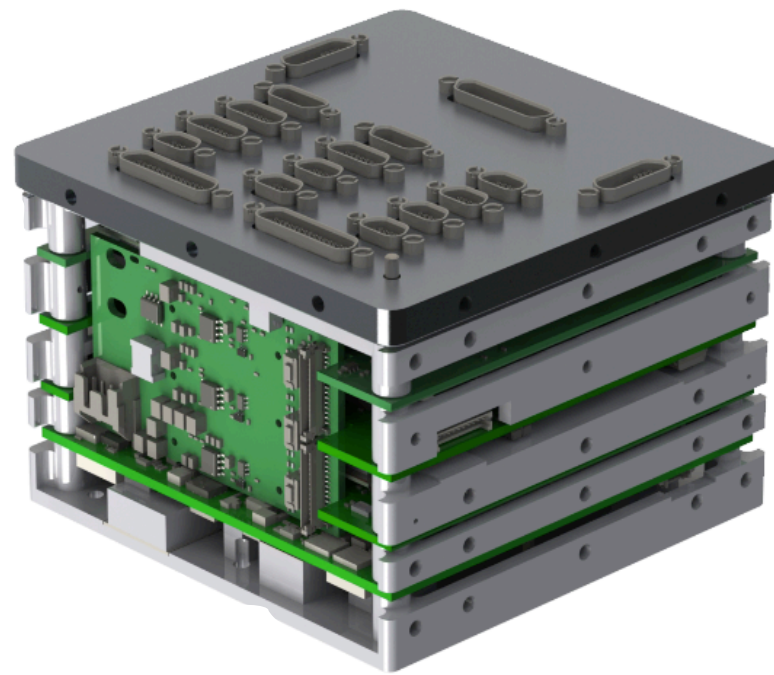


The Opportunity

The Opportunity

On-board Compute

Space-grade payload computers can run ML and image processing onboard.



Unibap iX10-101

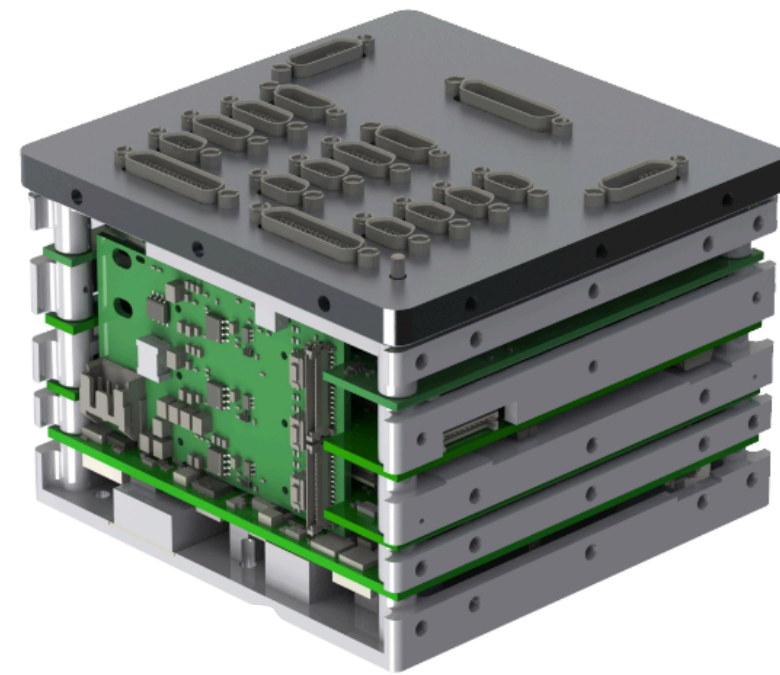


**NVIDIA Jetson
Orin NX**

The Opportunity

On-board Compute

Space-grade payload computers can run ML and image processing onboard.



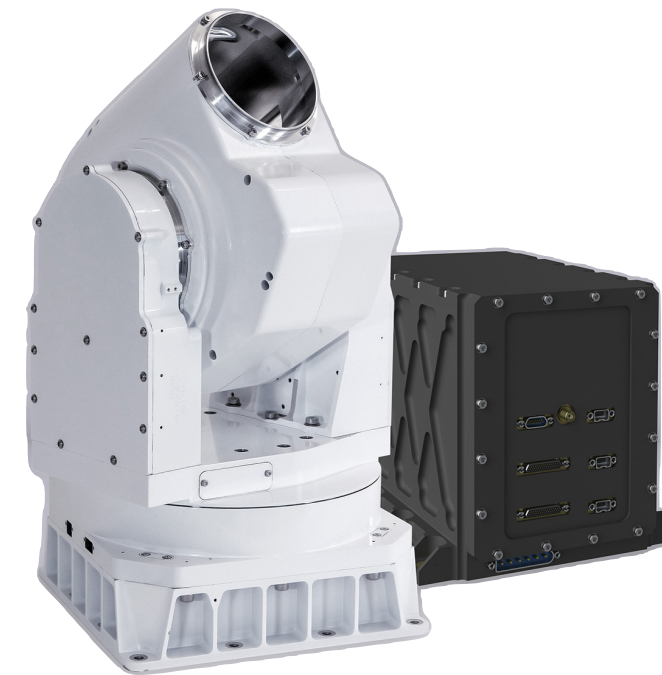
Unibap iX10-101



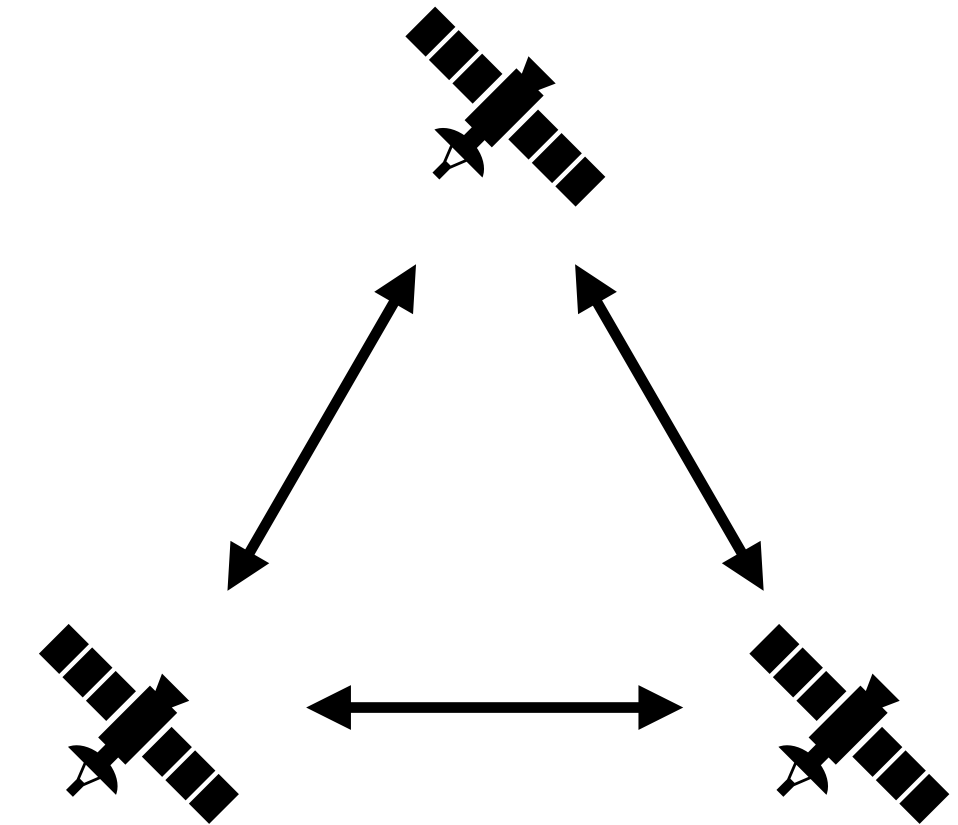
NVIDIA Jetson Orin NX

Inter-Satellite Communication

Satellites can communicate via high-speed optical inter-satellite links (ISLs)



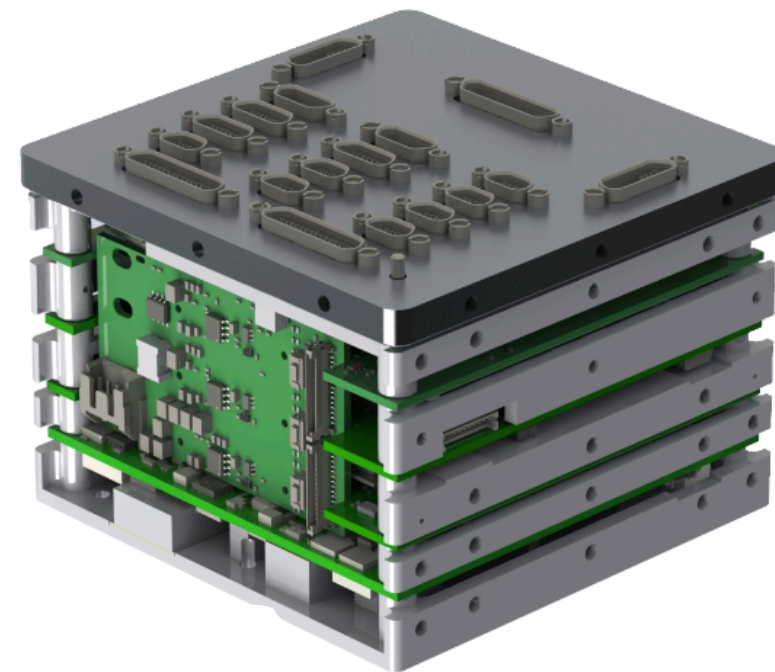
Condor Mk3 Terminal



The Opportunity

On-board Compute

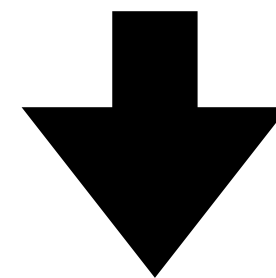
Space-grade payload computers can run ML and image processing onboard.



Unibap iX10-101



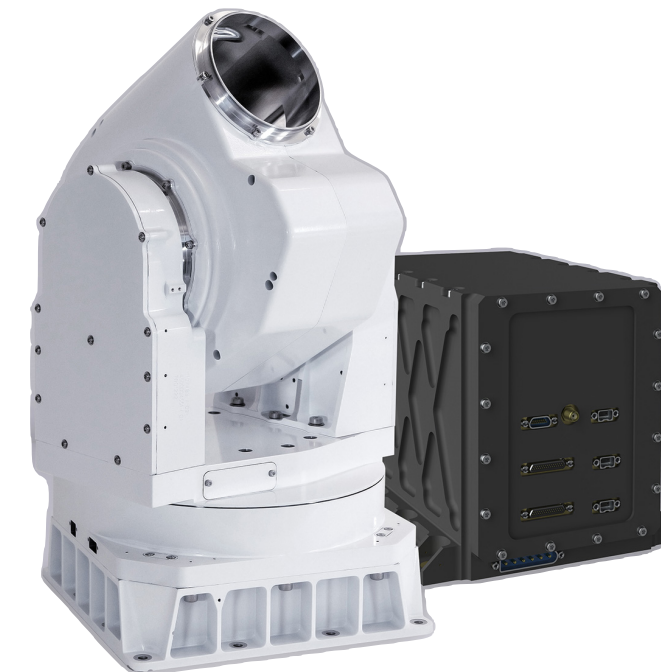
NVIDIA Jetson Orin NX



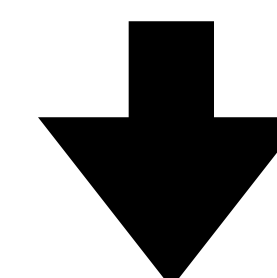
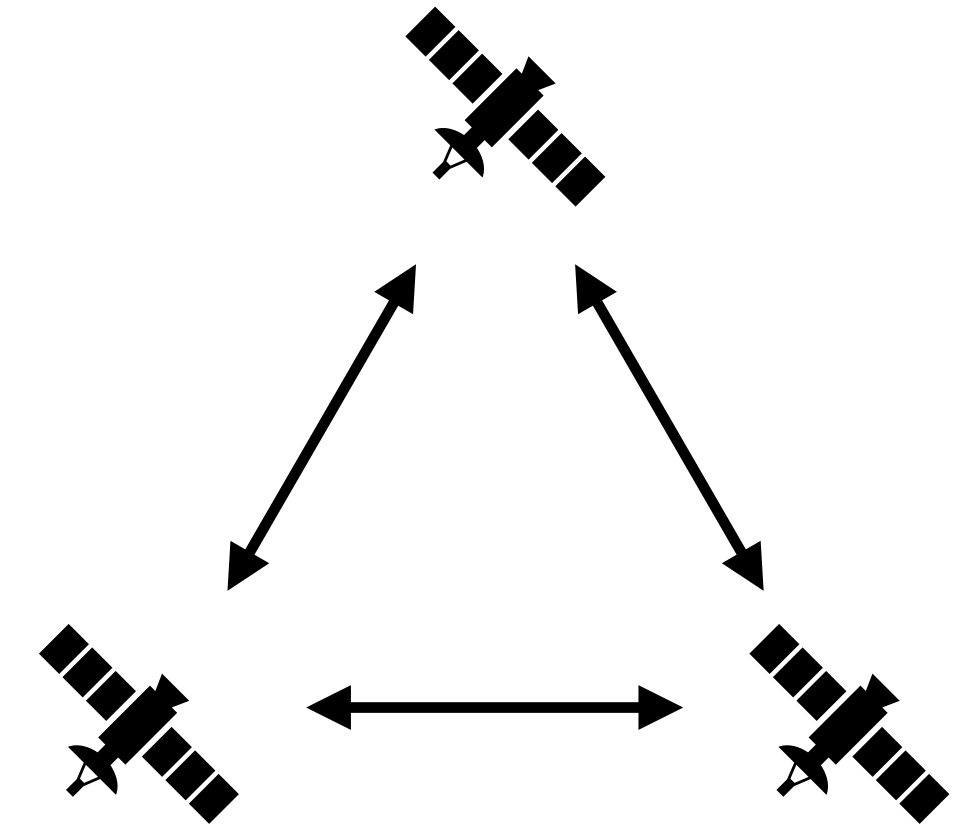
Process data in-orbit, where it is collected.

Inter-Satellite Communication

Satellites can communicate via high-speed optical inter-satellite links (ISLs)



Condor Mk3 Terminal

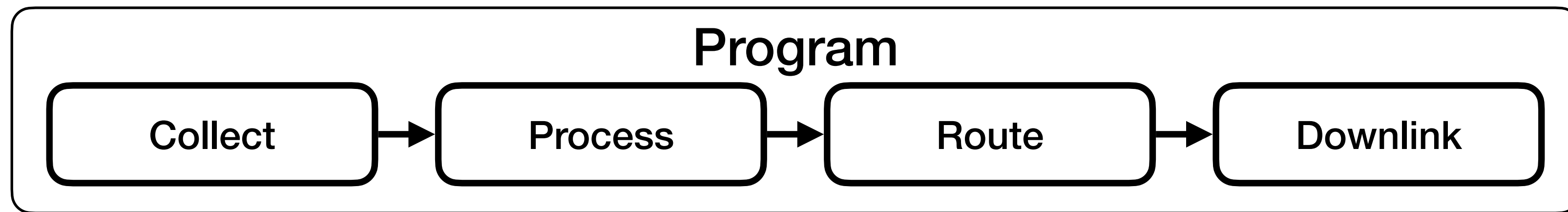


Route and downlink only the answer.

The Challenge

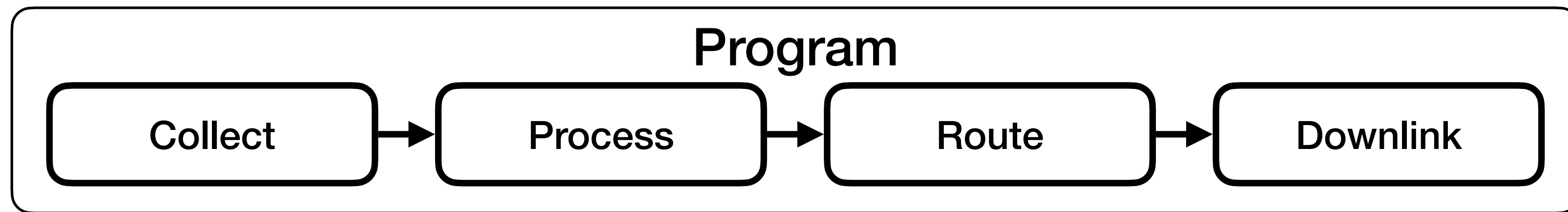
The Challenge

How do we express a live EO application as **one** program?



The Challenge

How do we express a live EO application as **one** program?

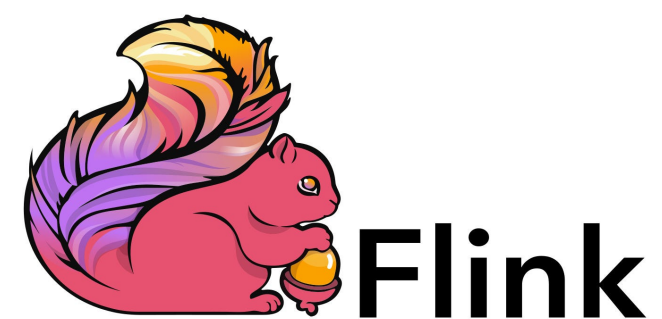
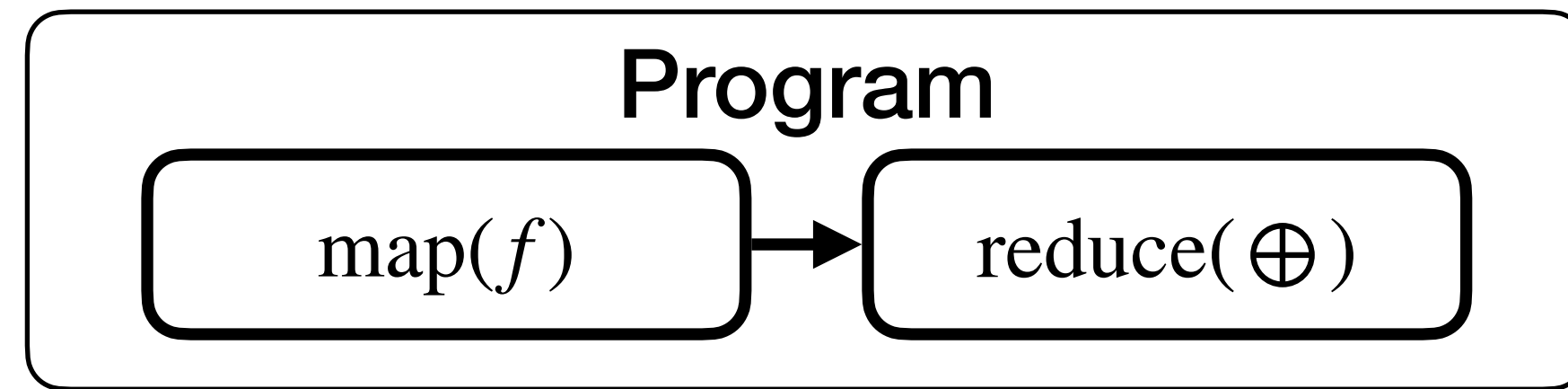


➡ Need a **programming model**

From Cloud to Space

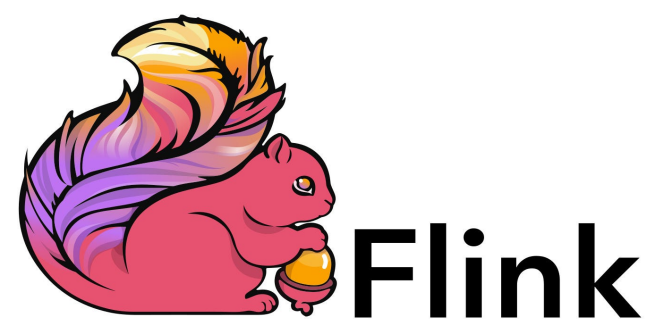
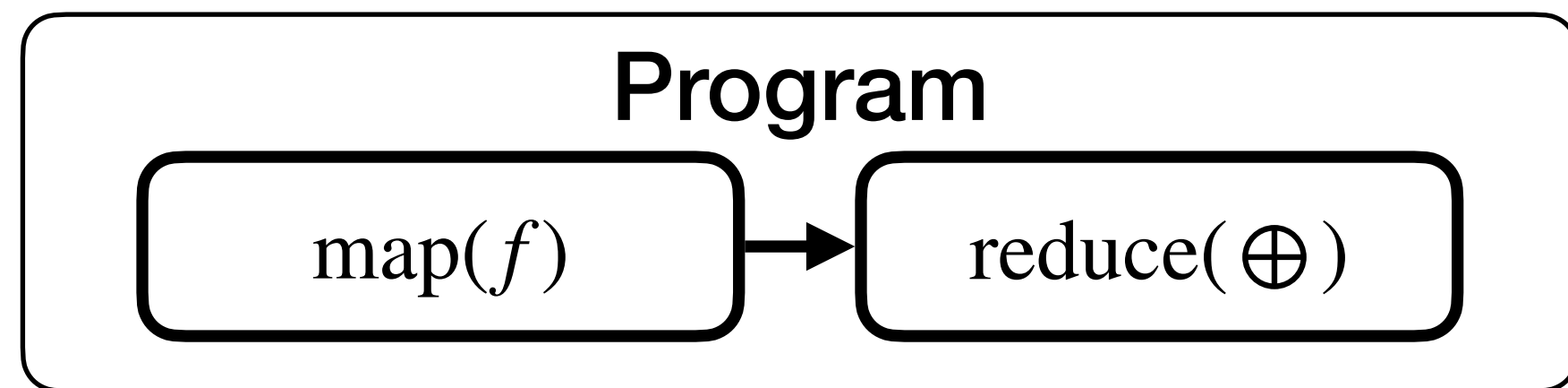
From Cloud to Space

Cloud Data Processing

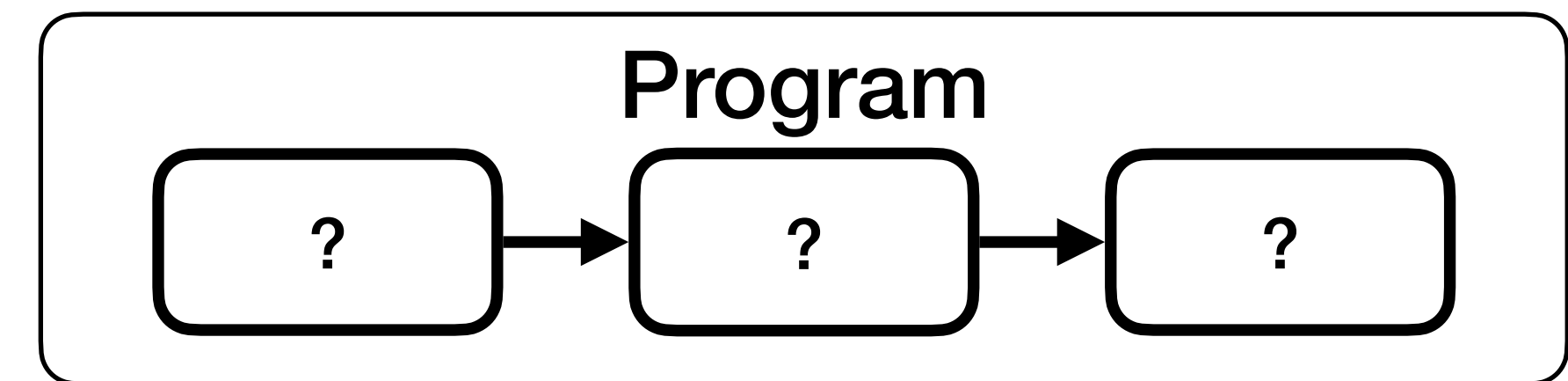


From Cloud to Space

Cloud Data Processing



Live EO in Space



Can live EO be made as simple to program as cloud data processing?

SpaceCoMP (Space-Collect-Map-Reduce Processing)

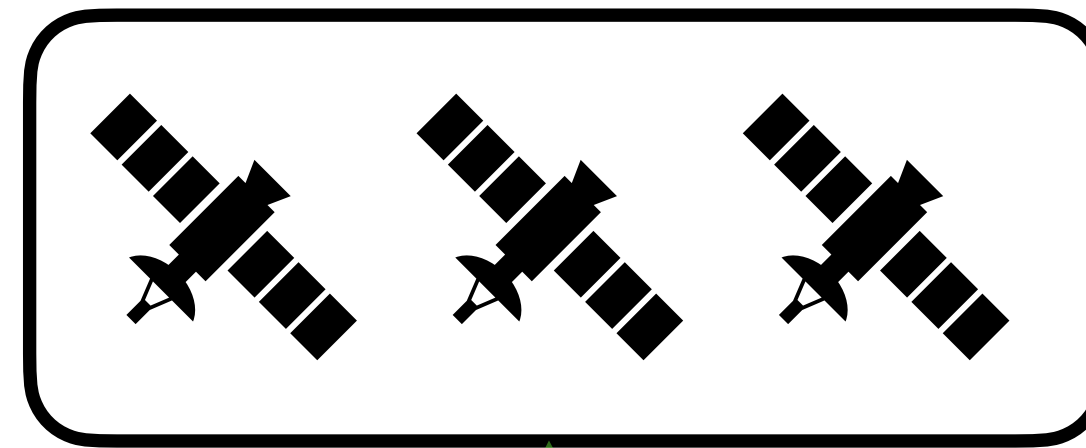
A MapReduce-inspired programming model for Earth Observation



SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation

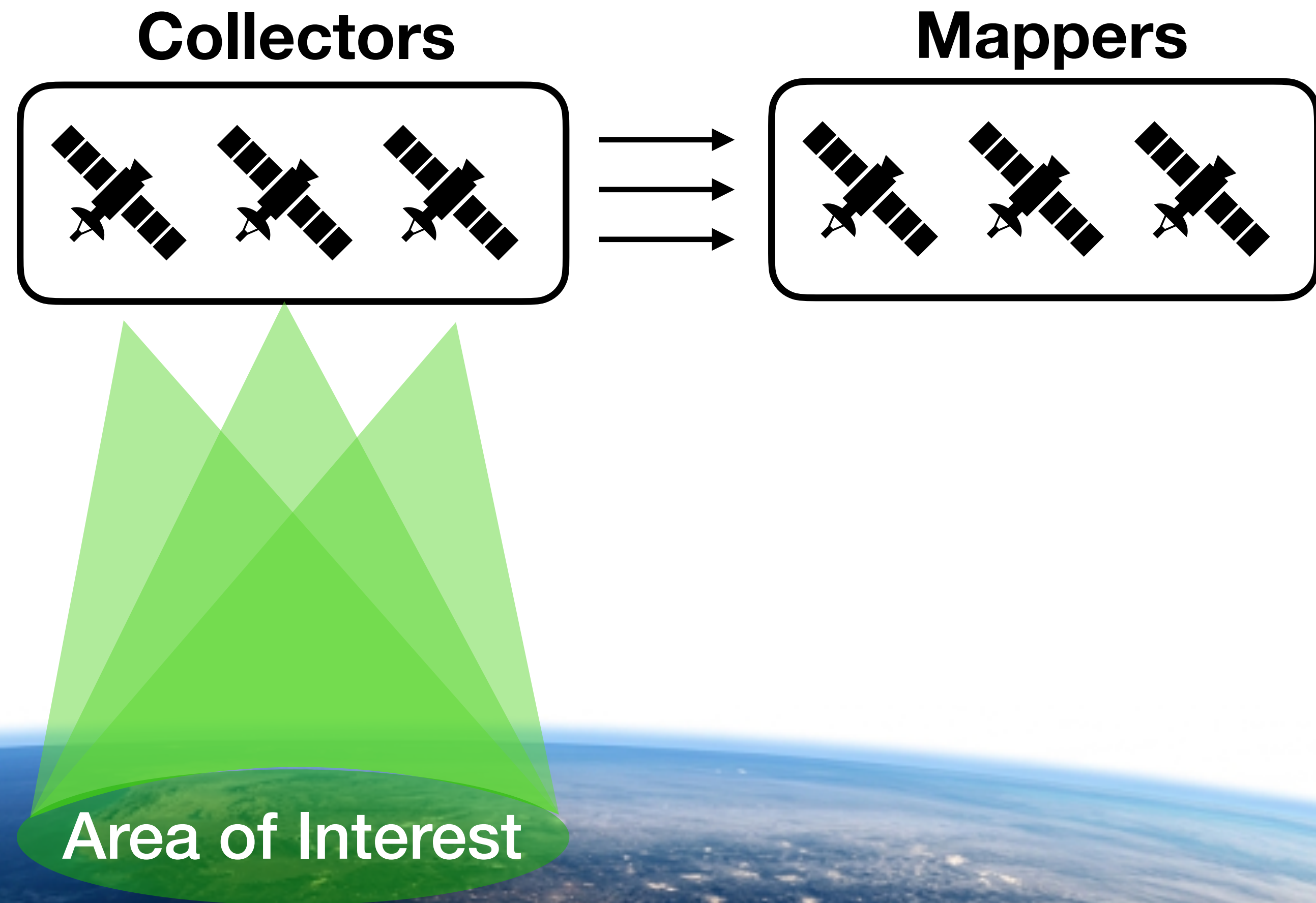
Collectors



Area of Interest

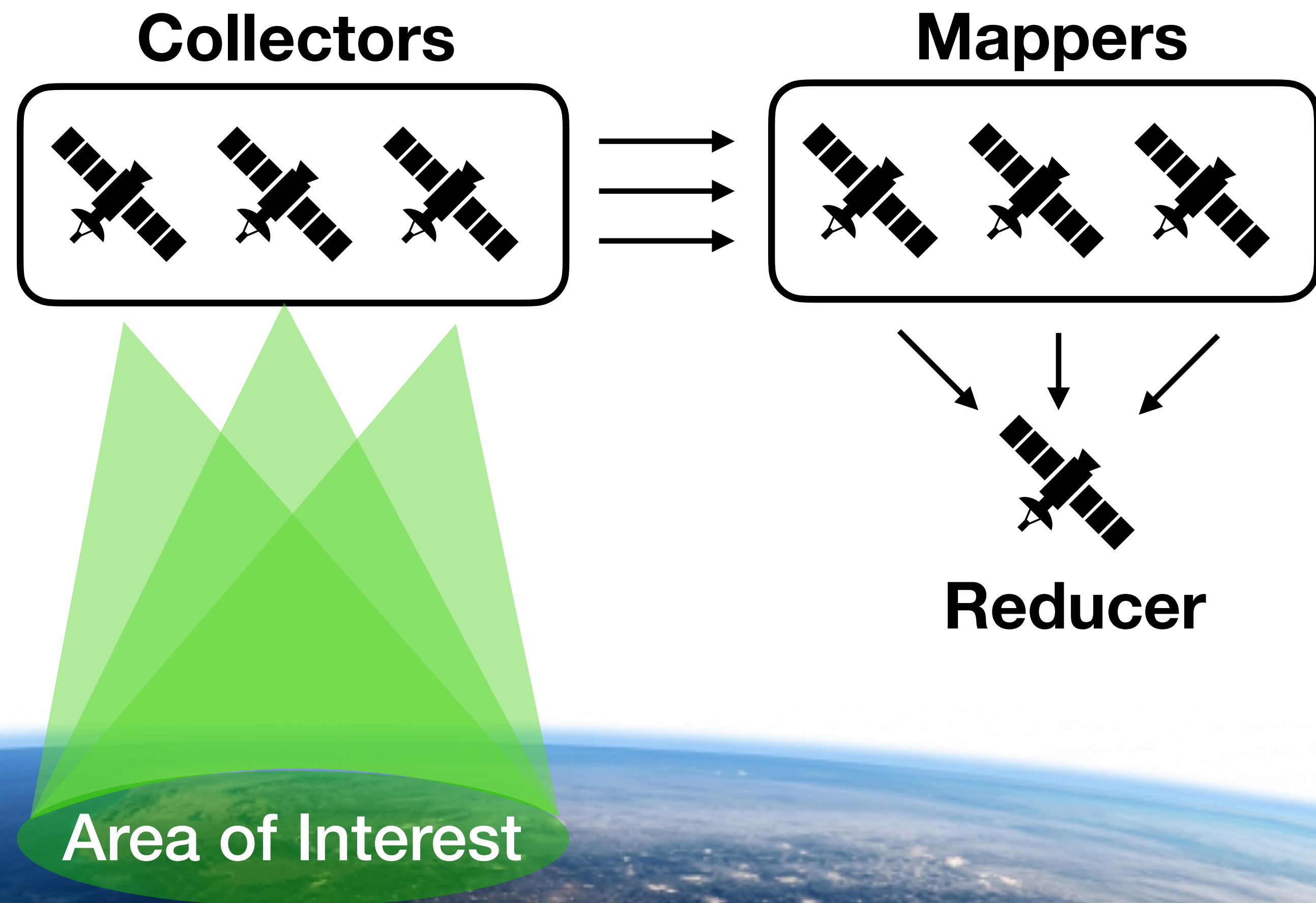
SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation



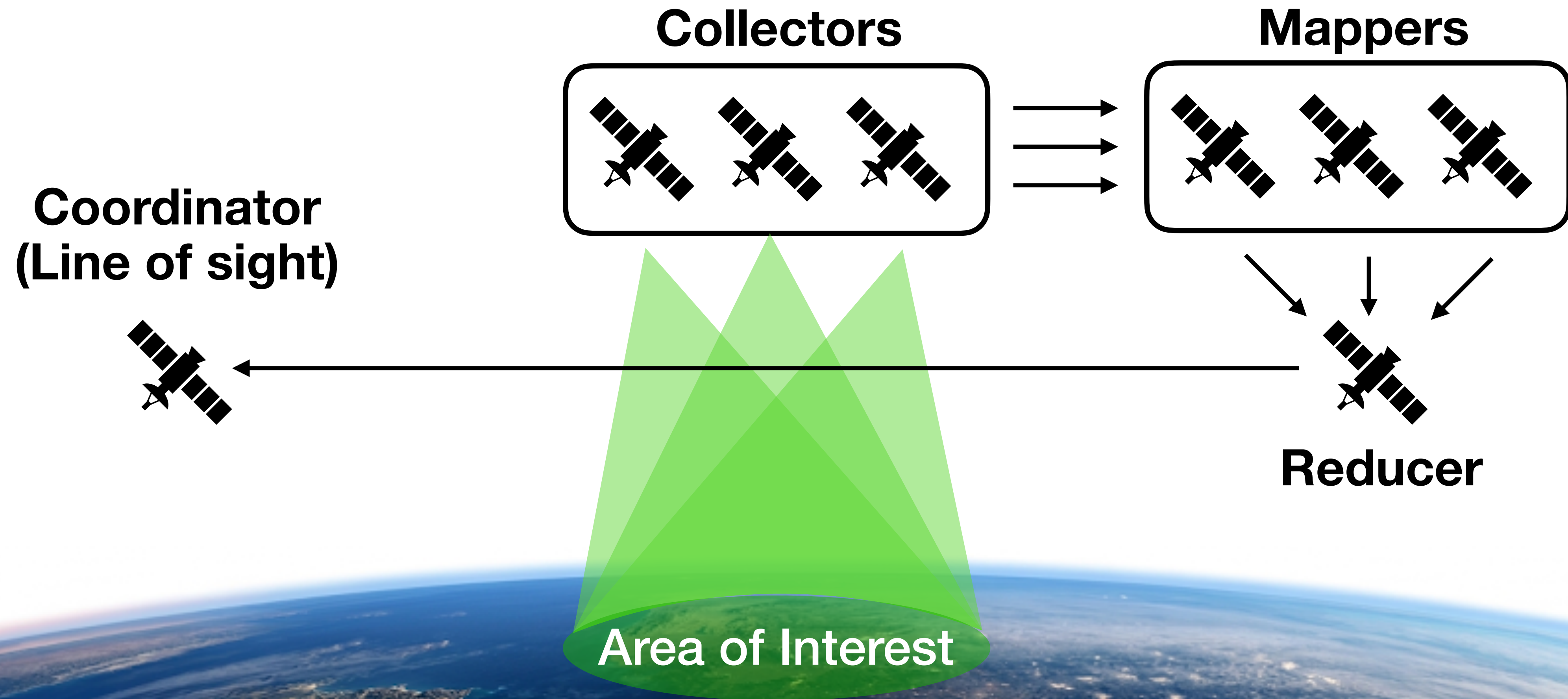
SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation



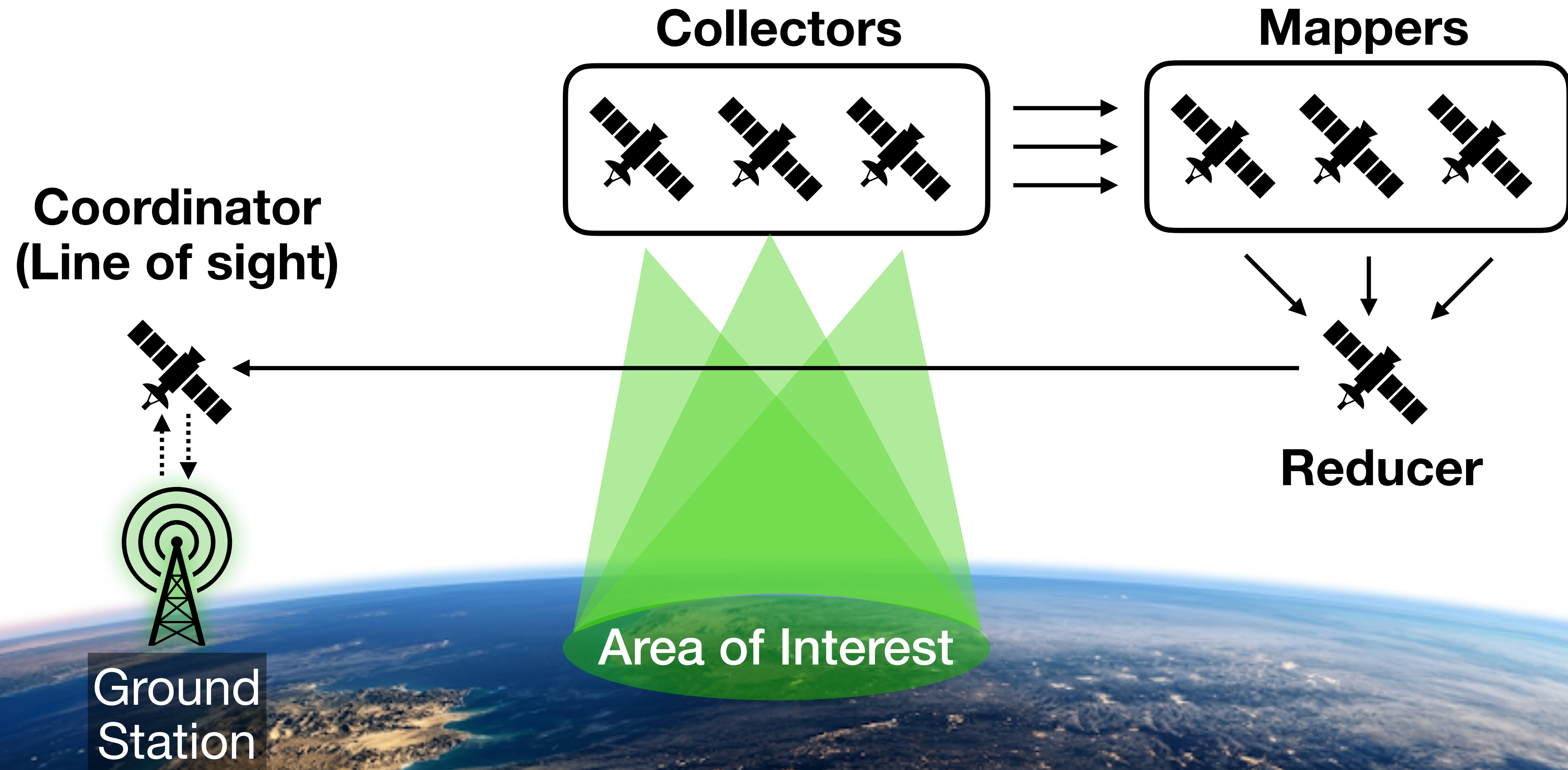
SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation



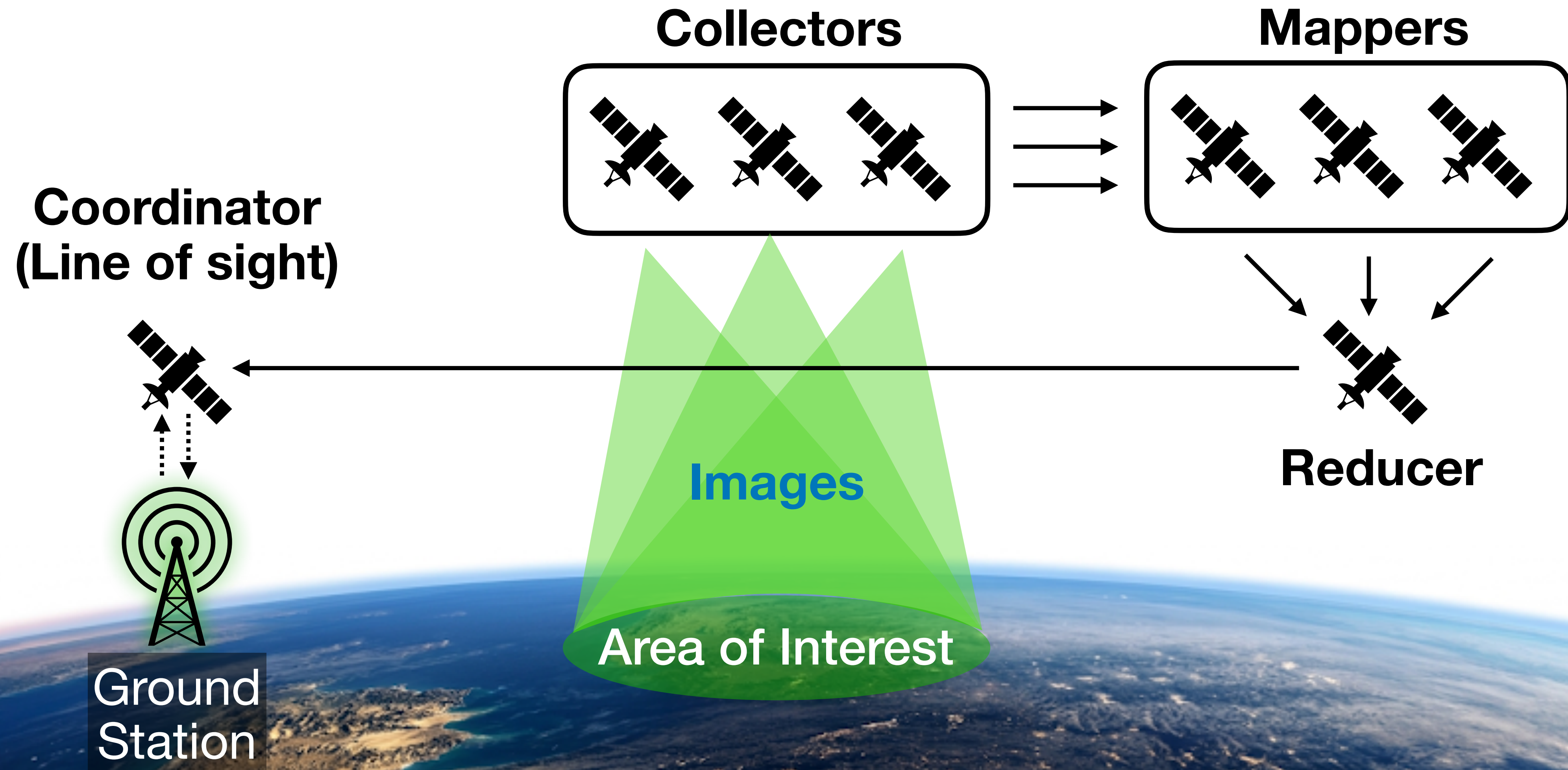
SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation



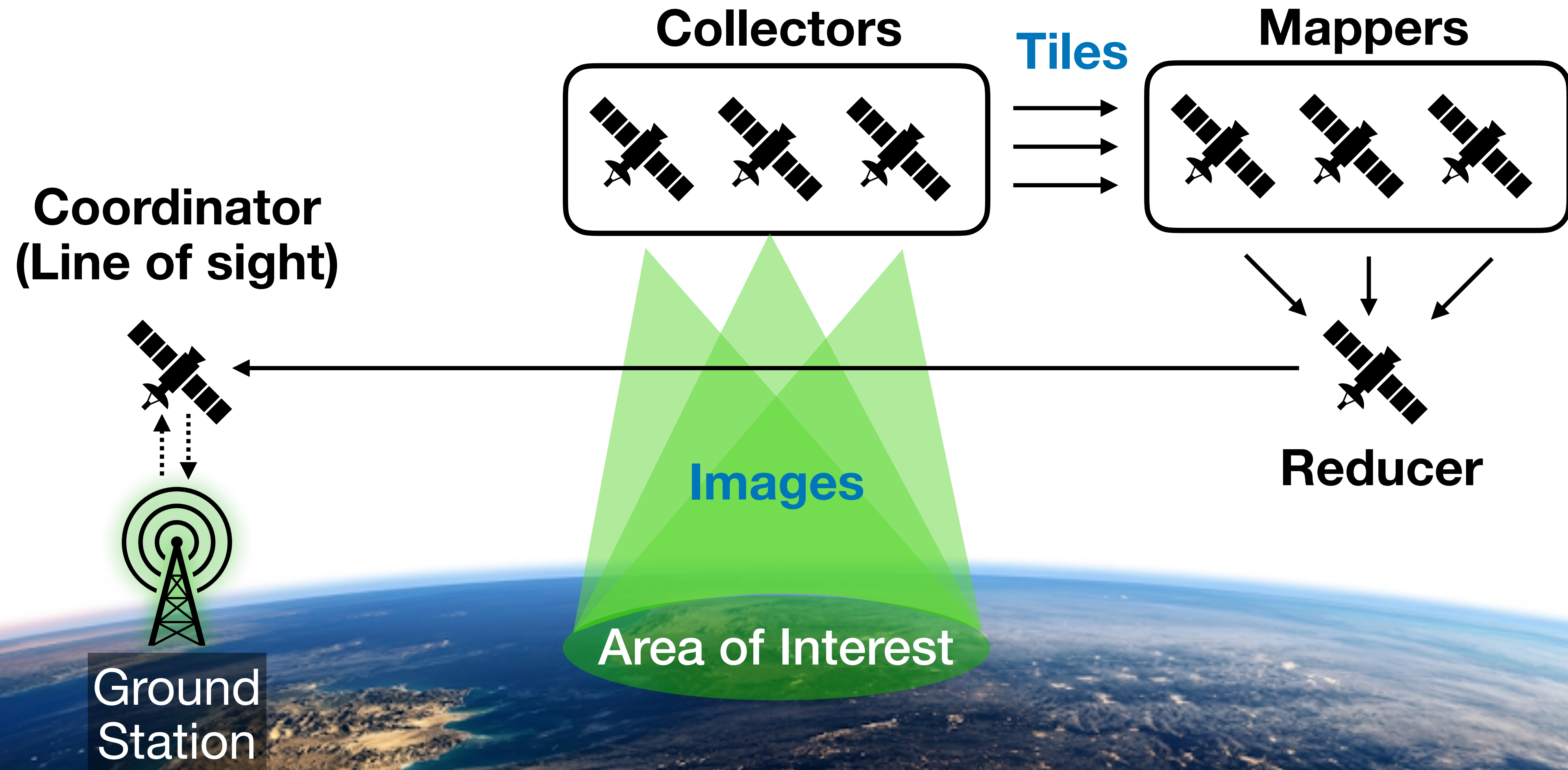
SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation



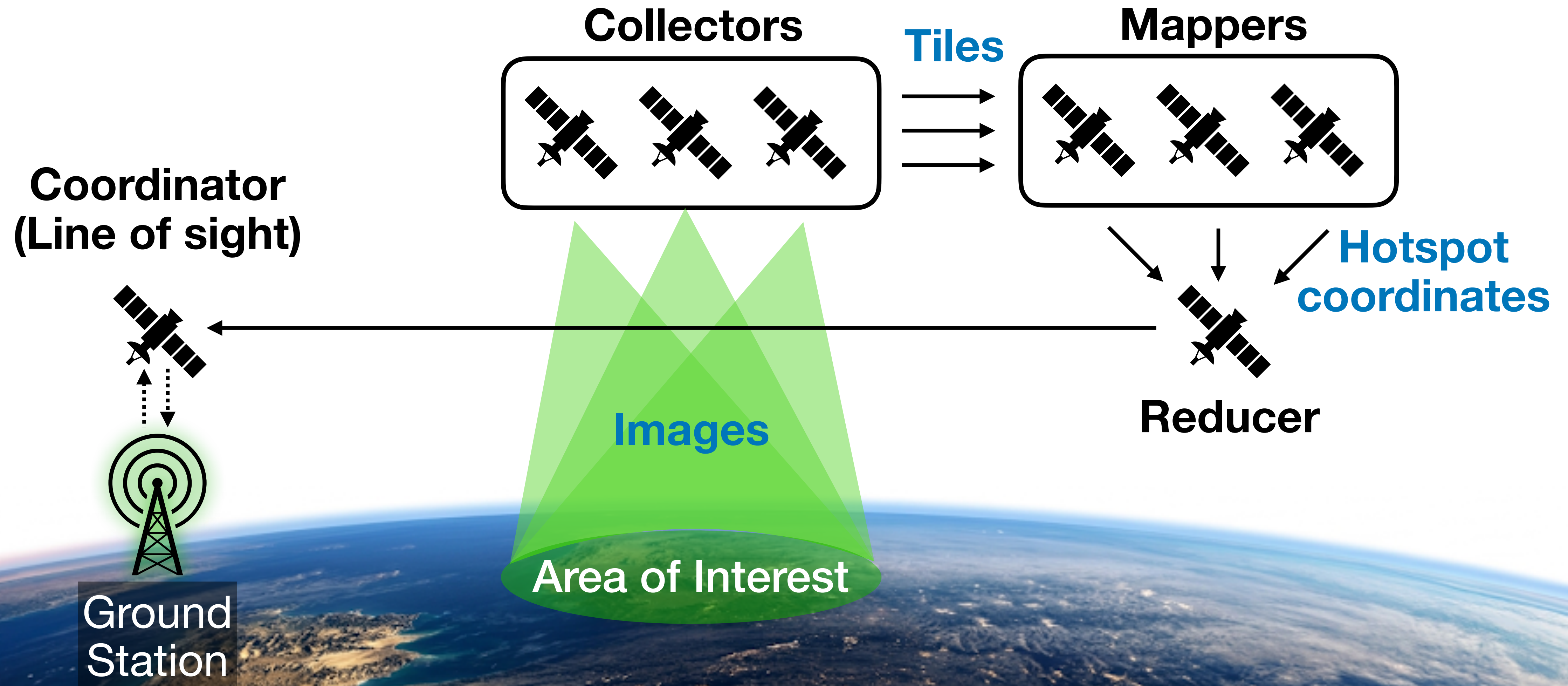
SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation



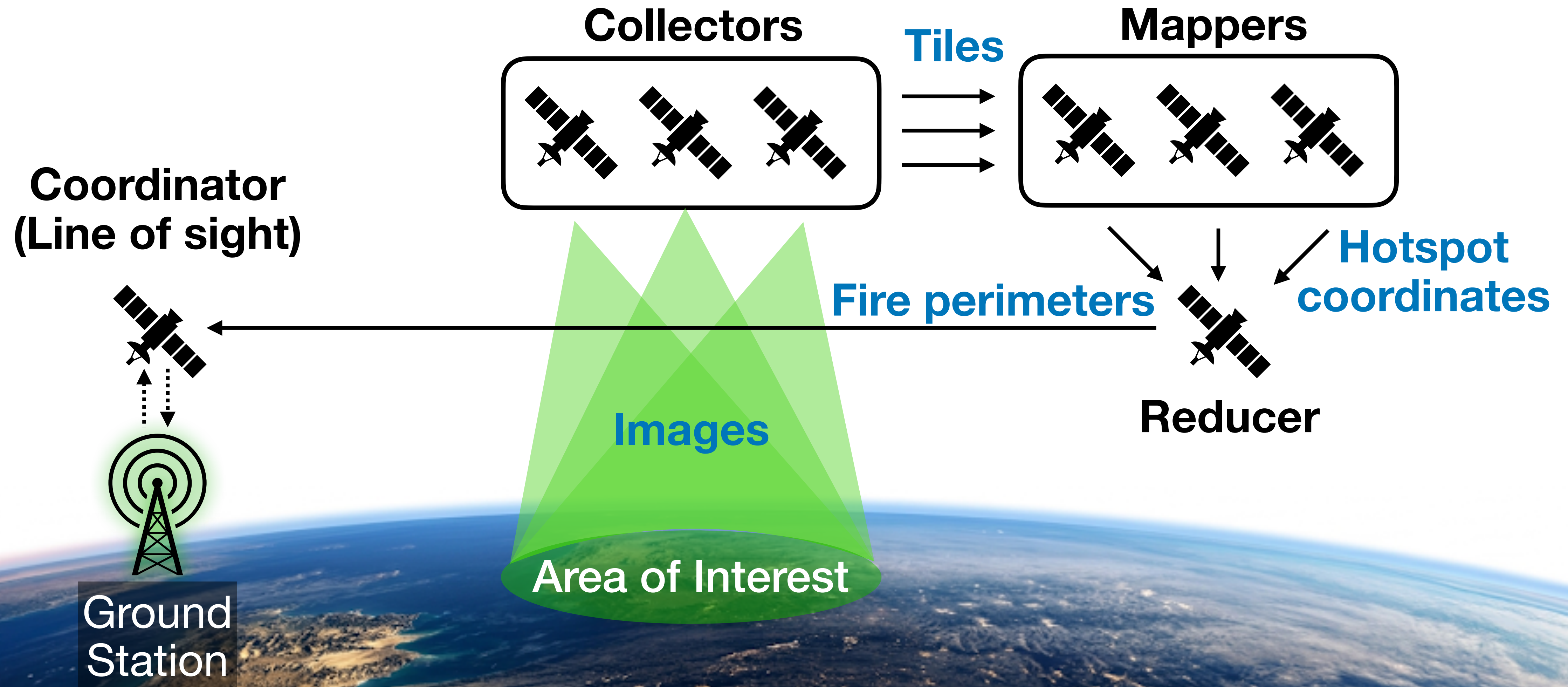
SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation



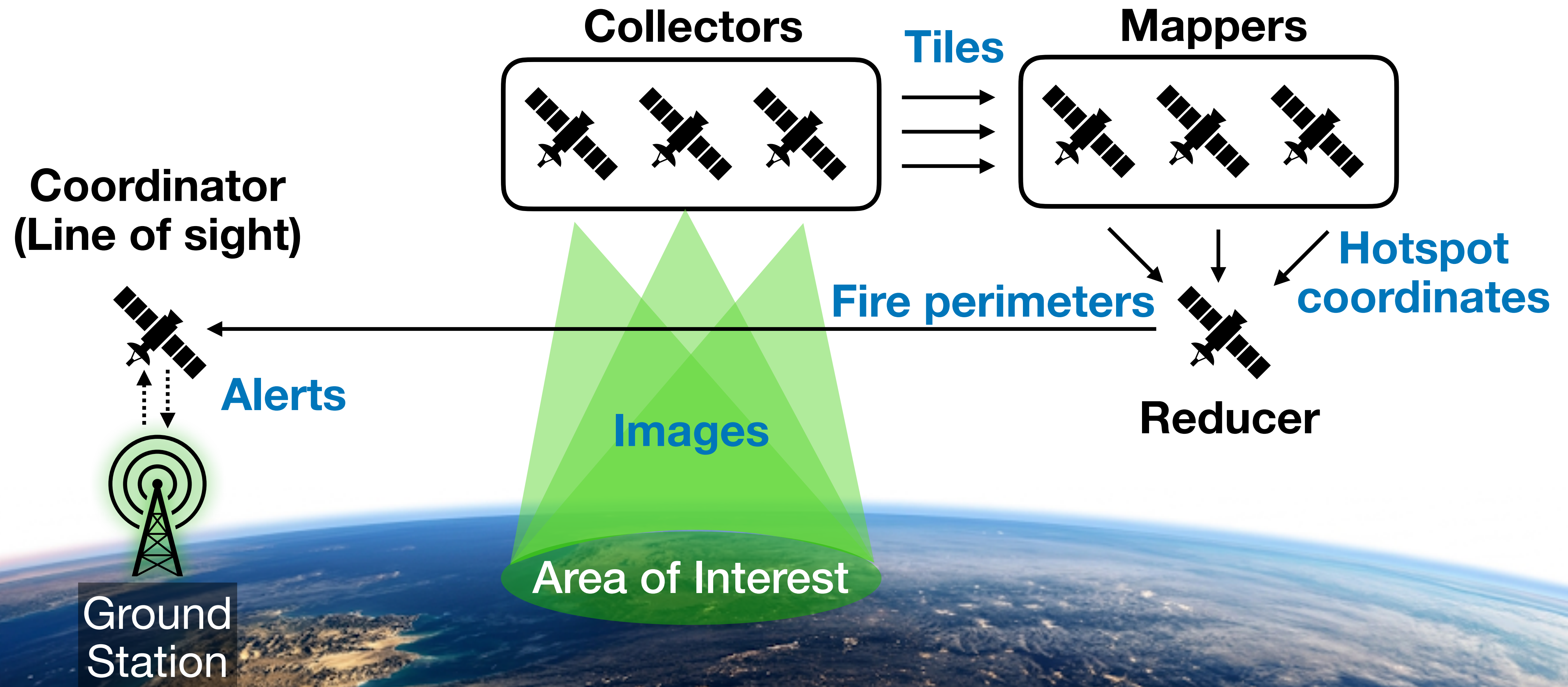
SpaceCoMP (Space-Collect-Map-Reduce Processing)

A MapReduce-inspired programming model for Earth Observation



SpaceCoMP (Space-Collect-Map-Reduce Processing)

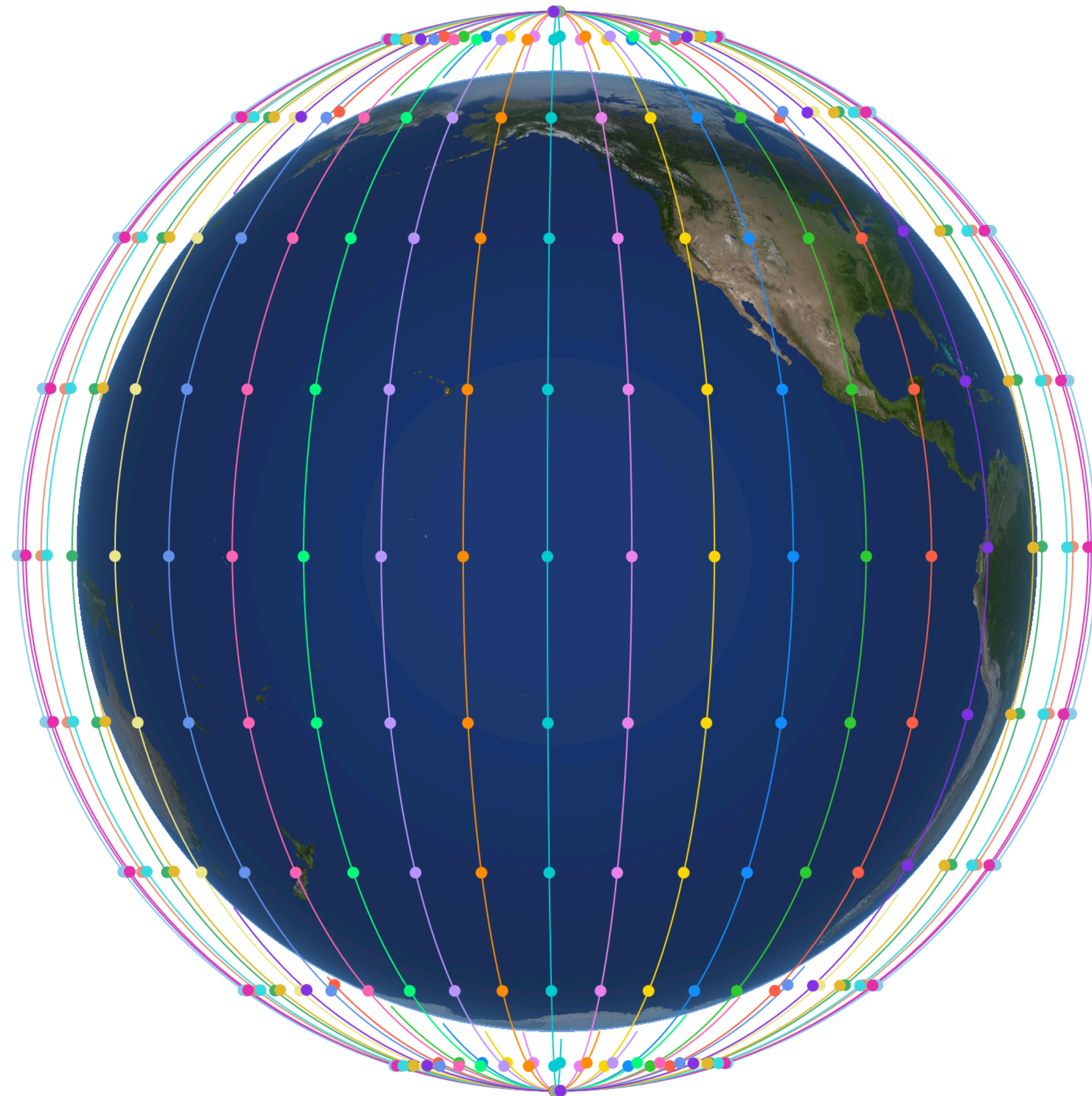
A MapReduce-inspired programming model for Earth Observation



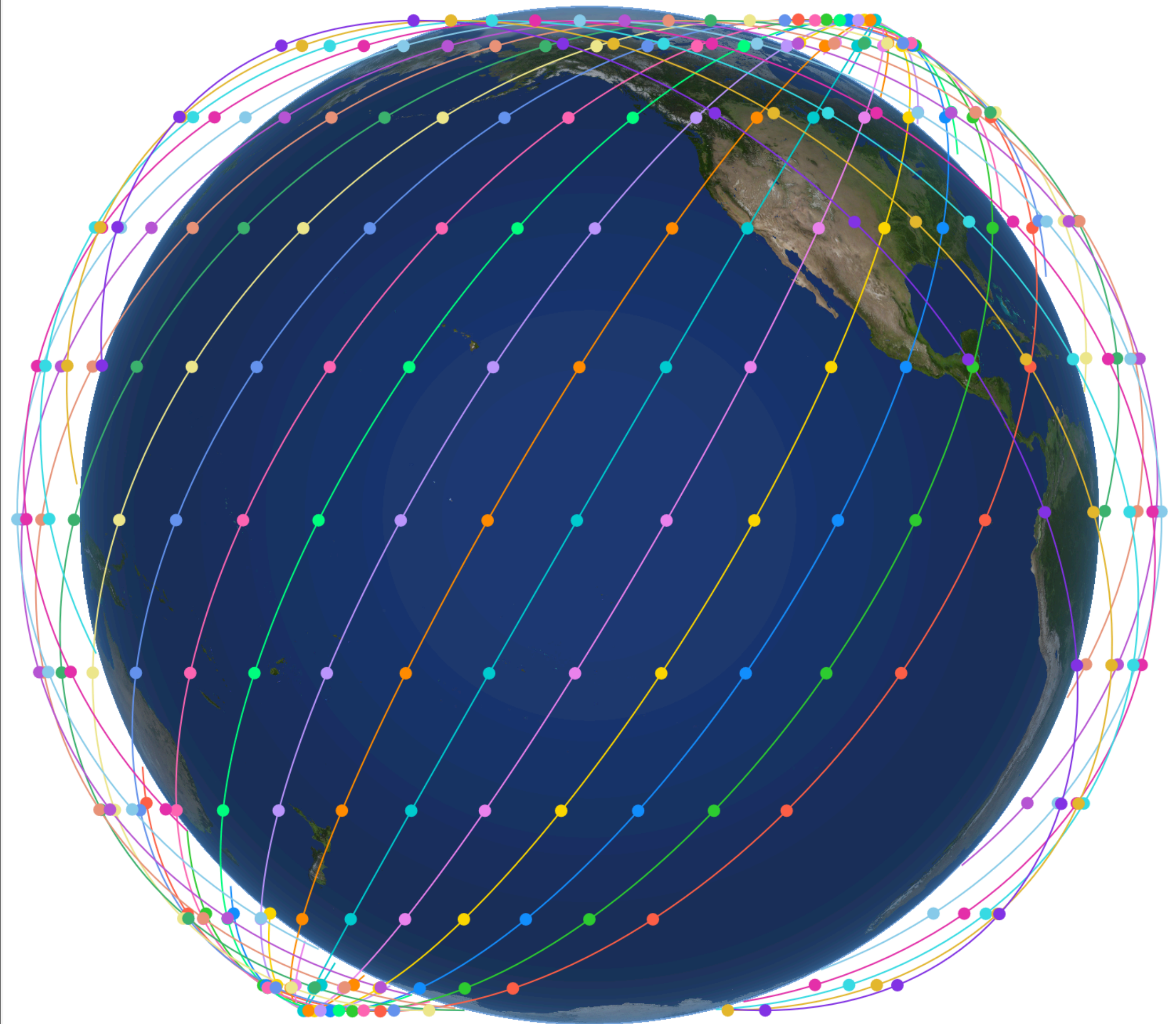
How does it work?

Orbits and Inclination

90° inclination

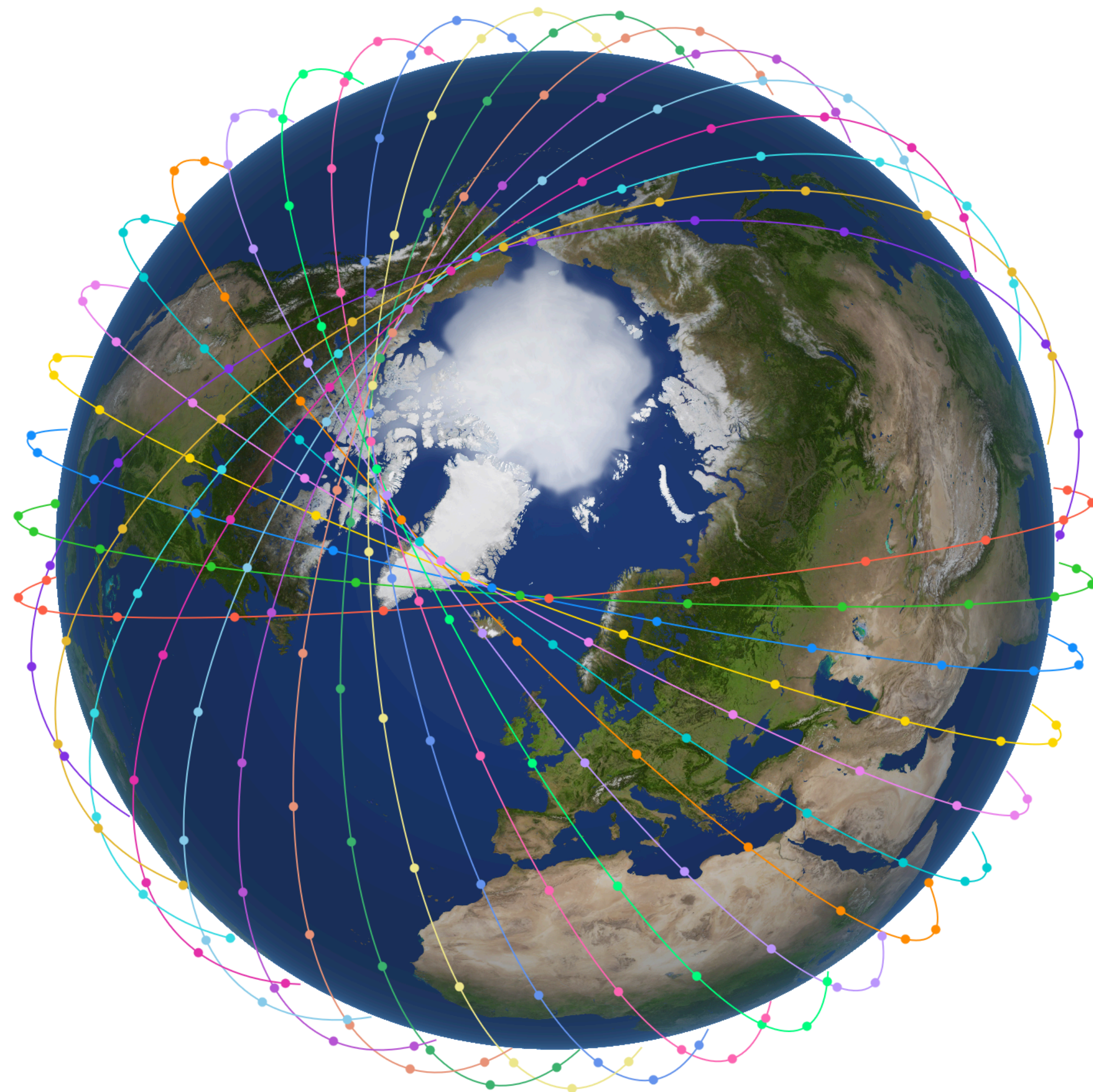


60° inclination

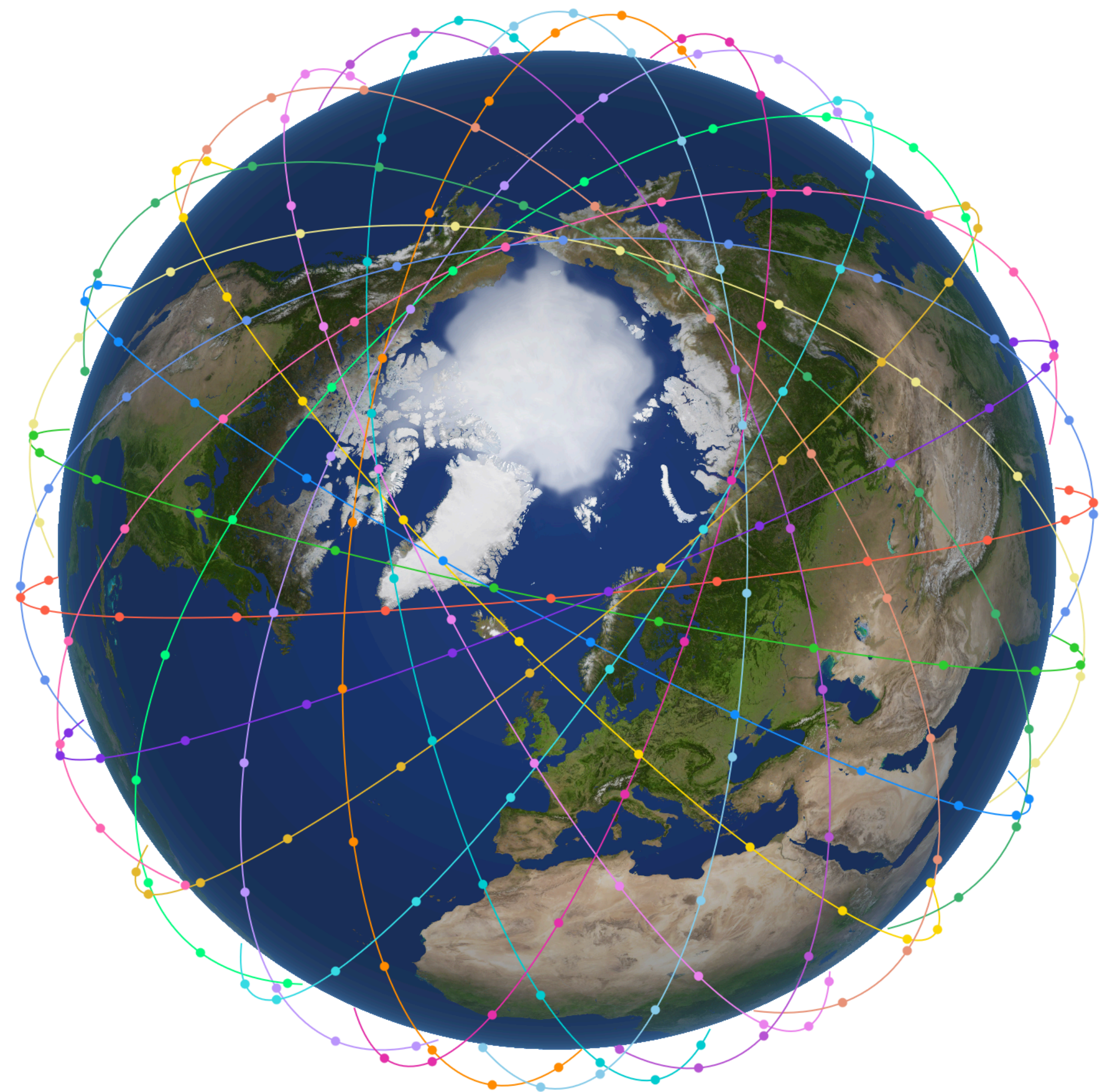


Walker Constellations

Walker Star

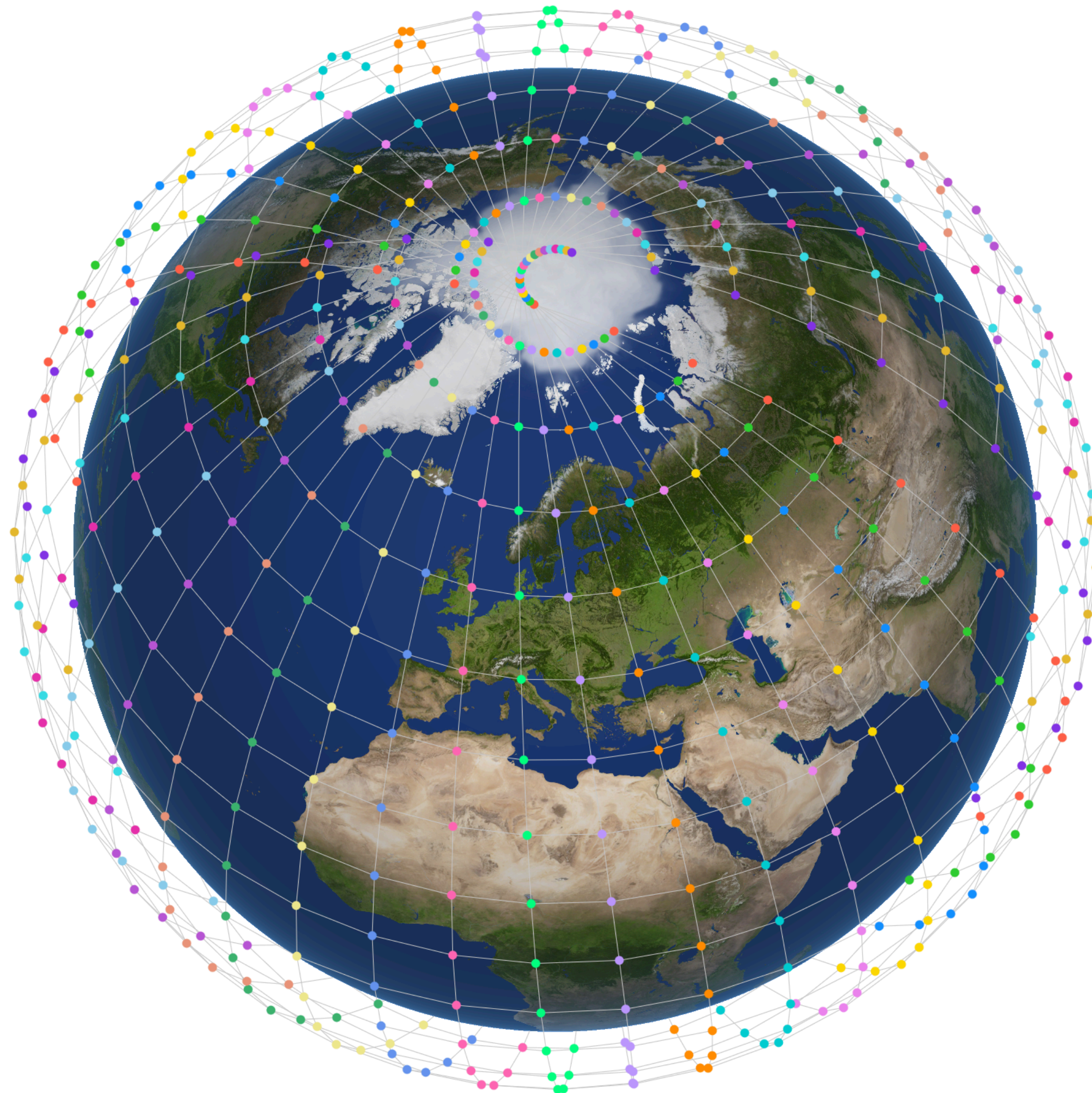


Walker Delta

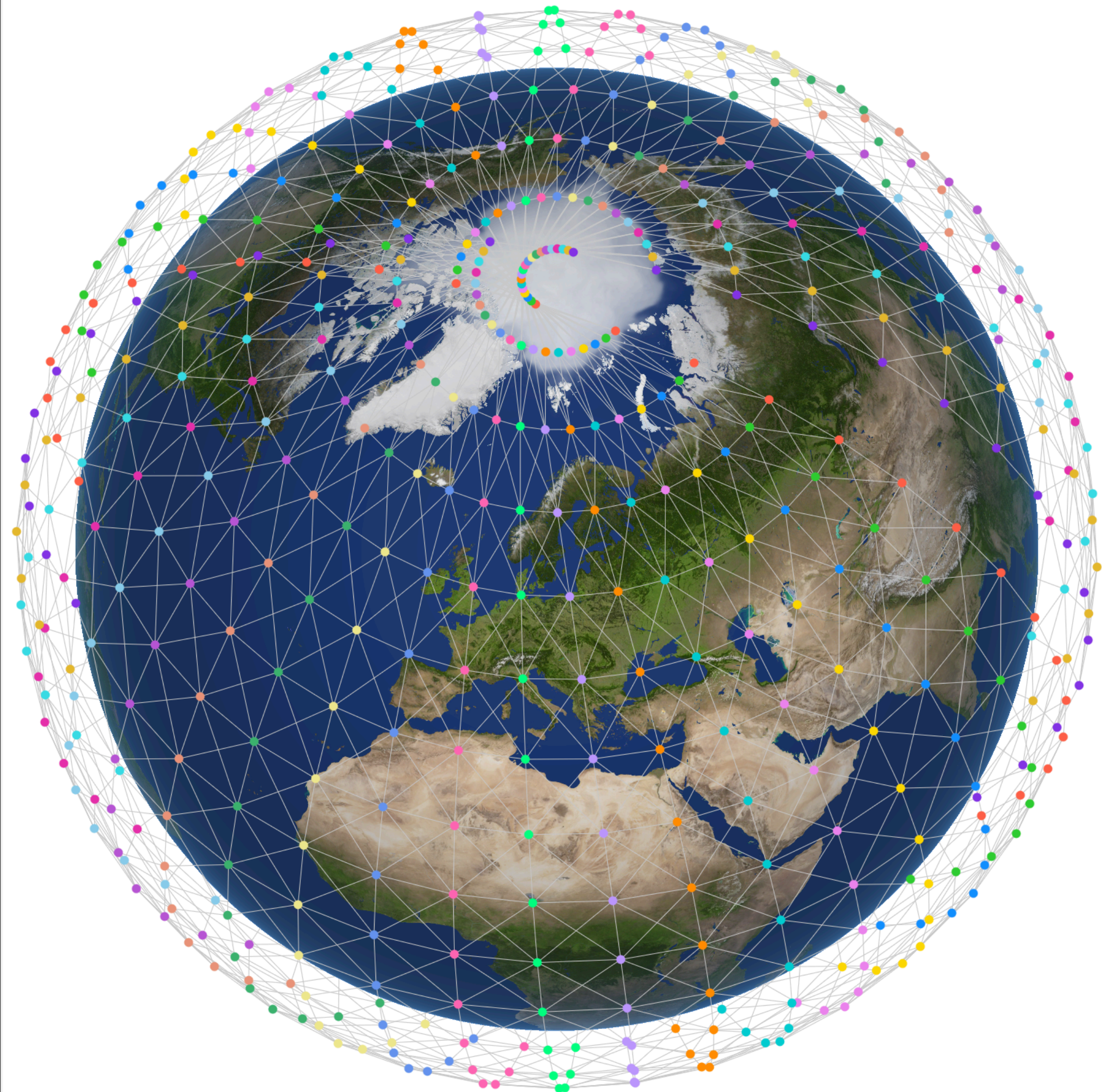


Inter-Satellite Links

4 ISLs per satellite (+Grid)

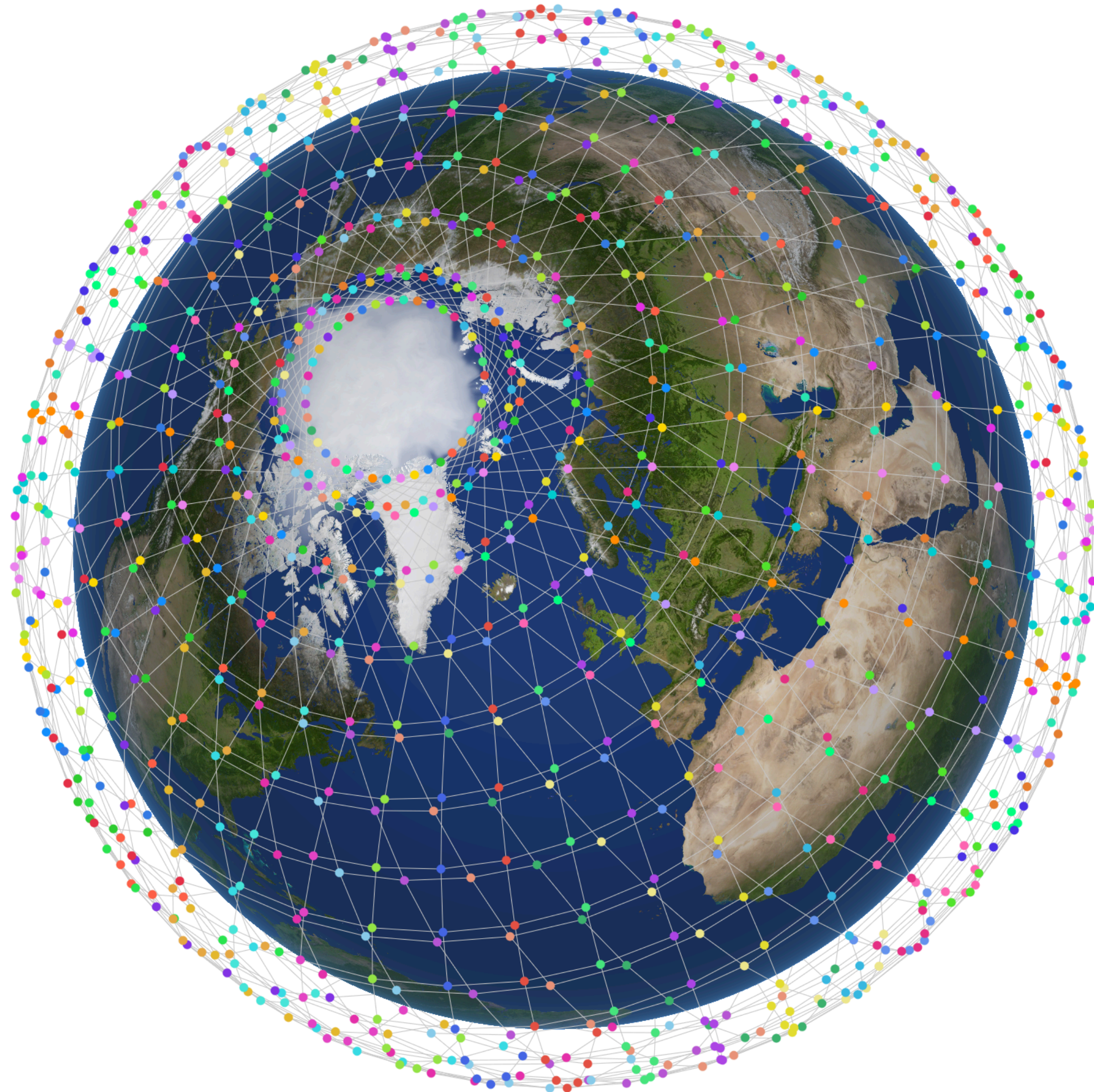


8 ISLs per satellite

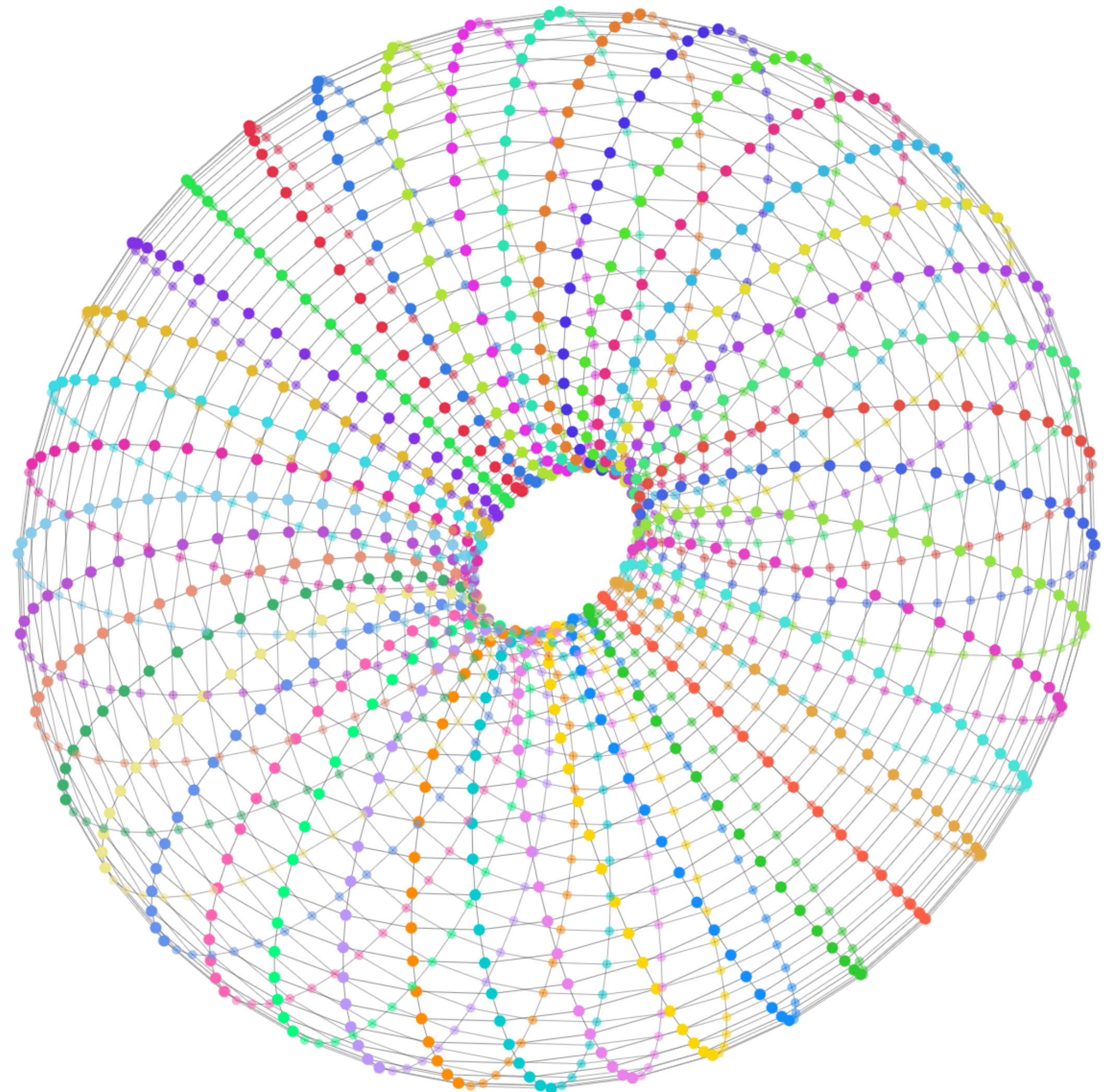


+Grid and Torus

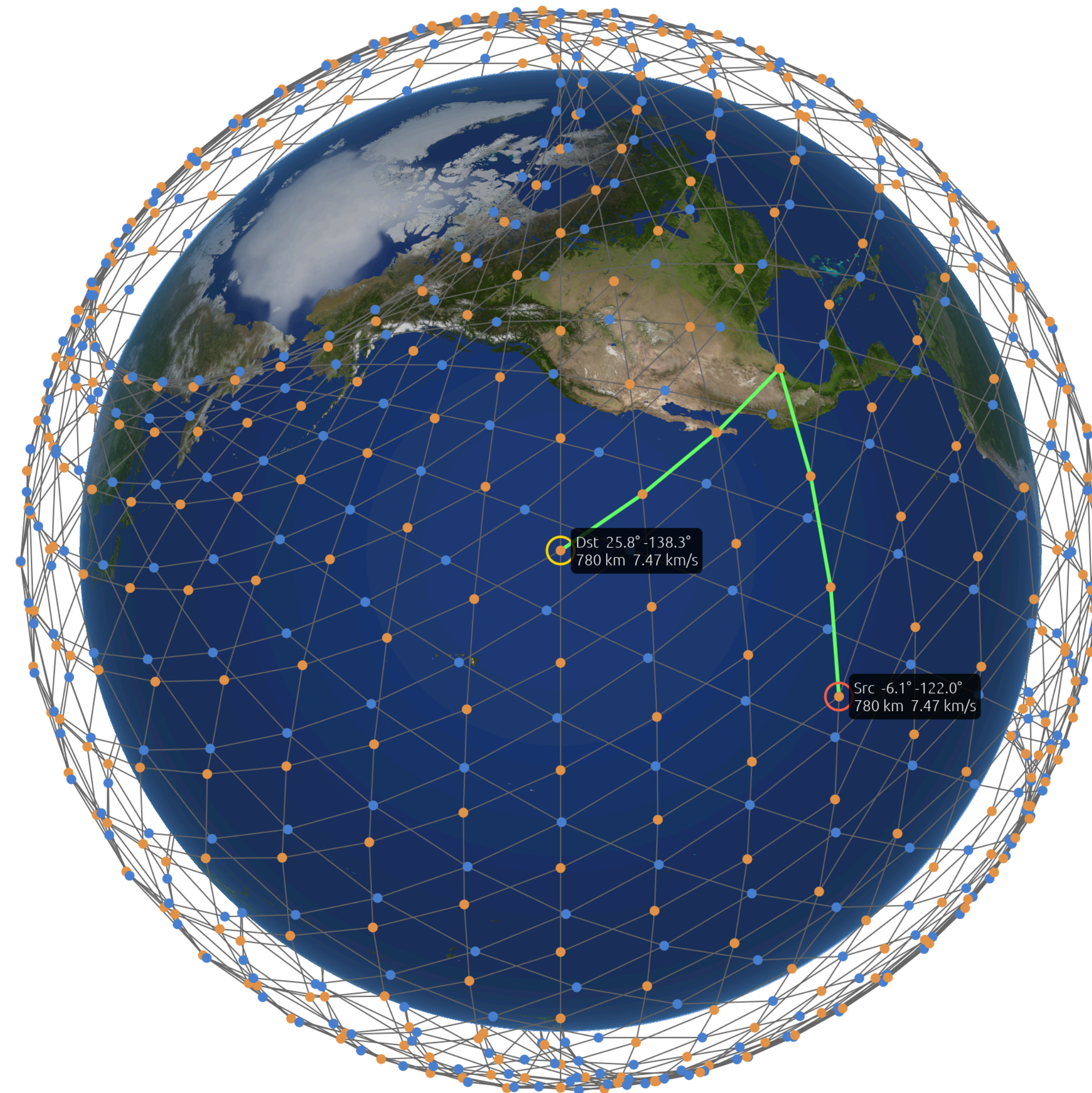
+Grid



Torus view

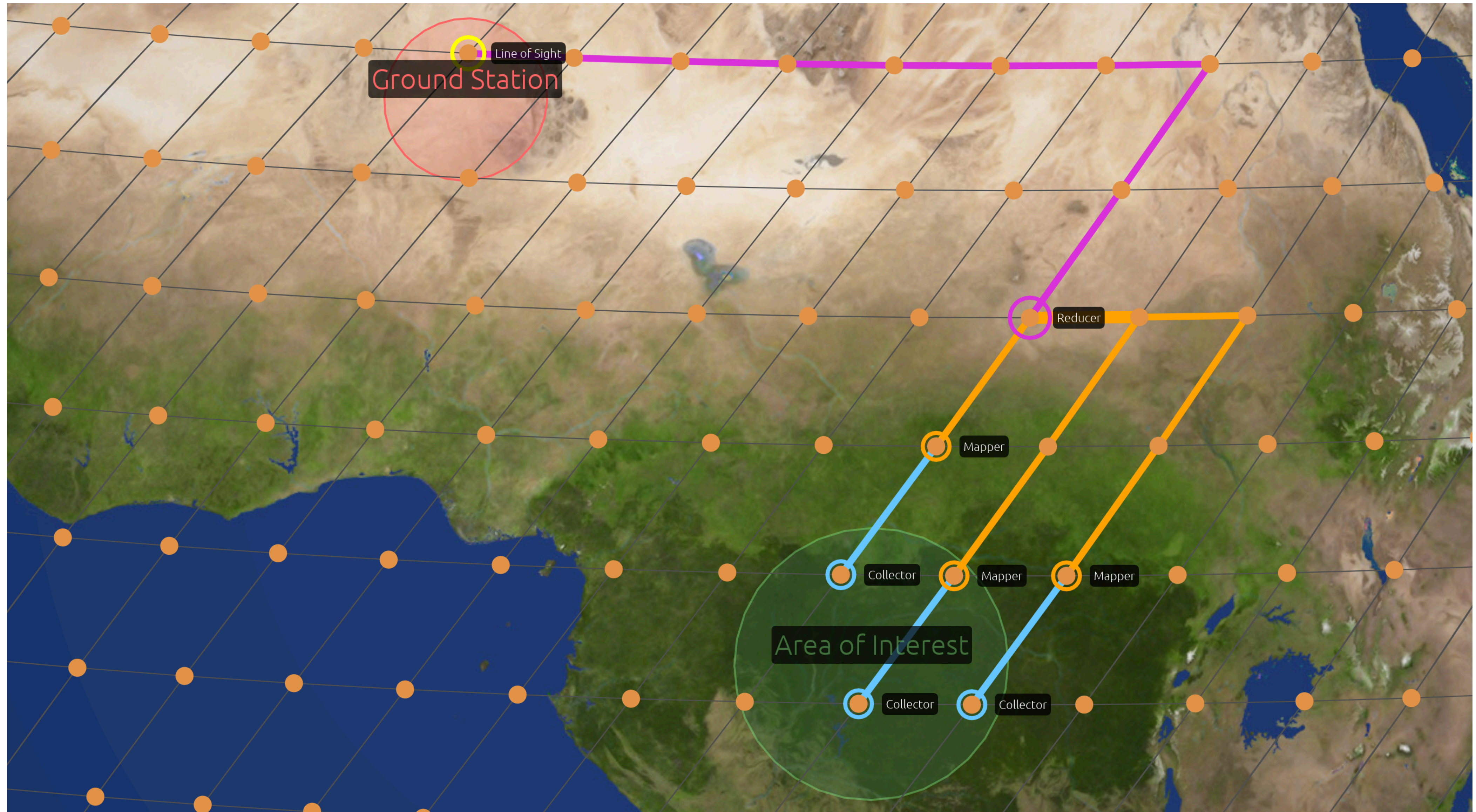


ISL Routing



Orange = Ascending
Blue = Descending

SpaceCoMP



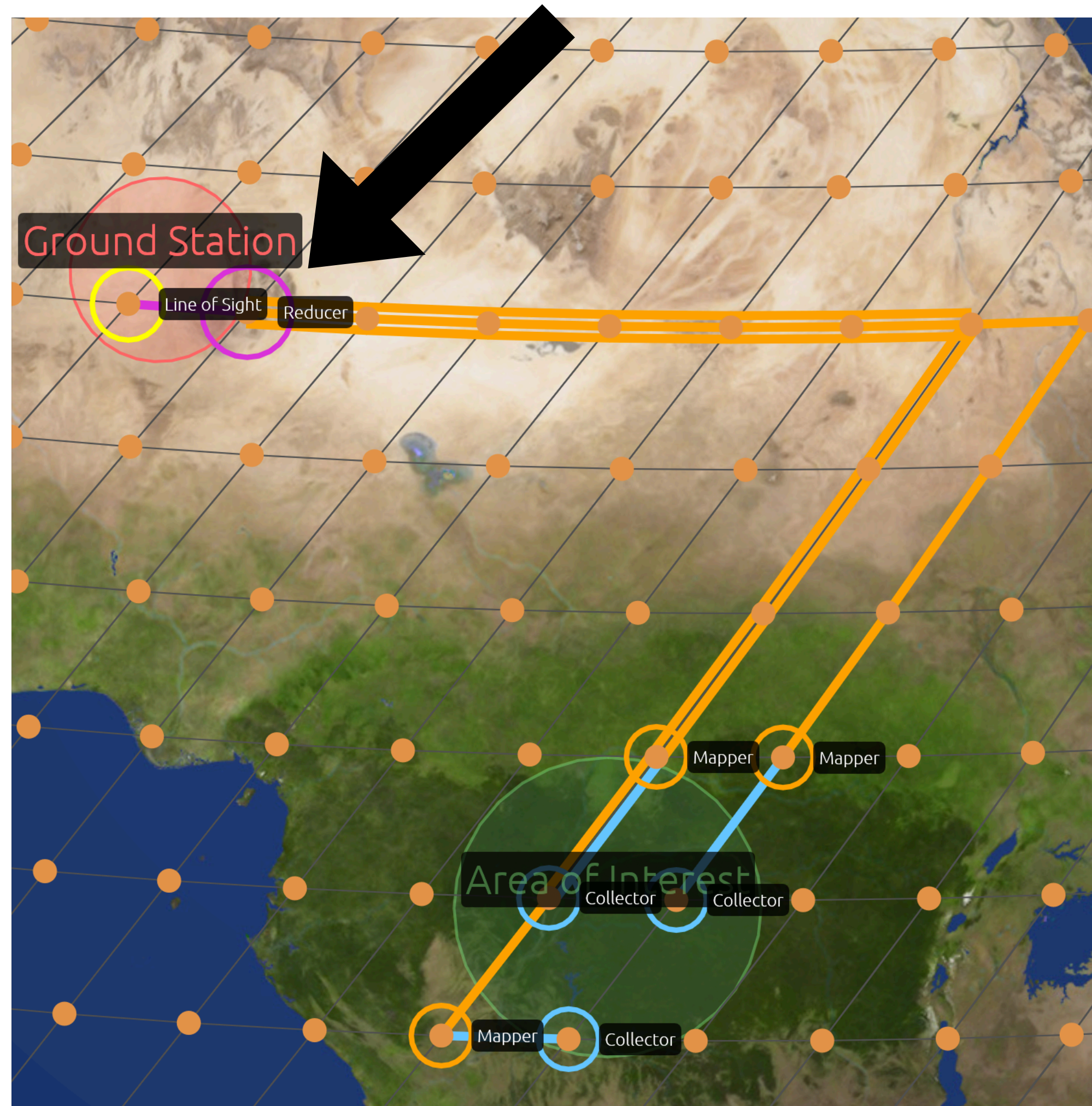
Optimisations

Evaluated in simulated constellations of 1k-10k satellites

Optimisation 1 - Reducer placement

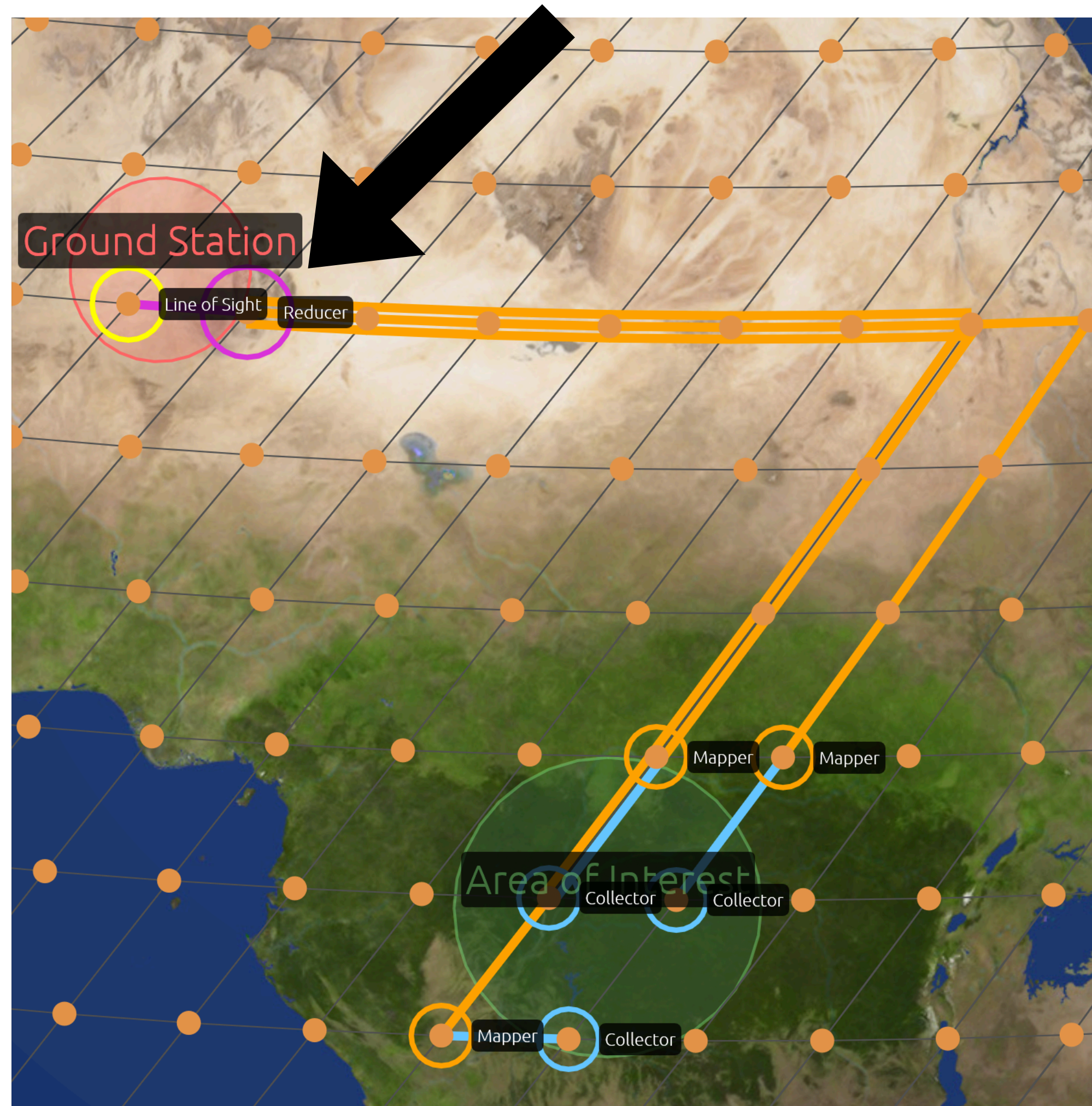
Optimisation 1 - Reducer placement

Closest to Line of sight (naïve)

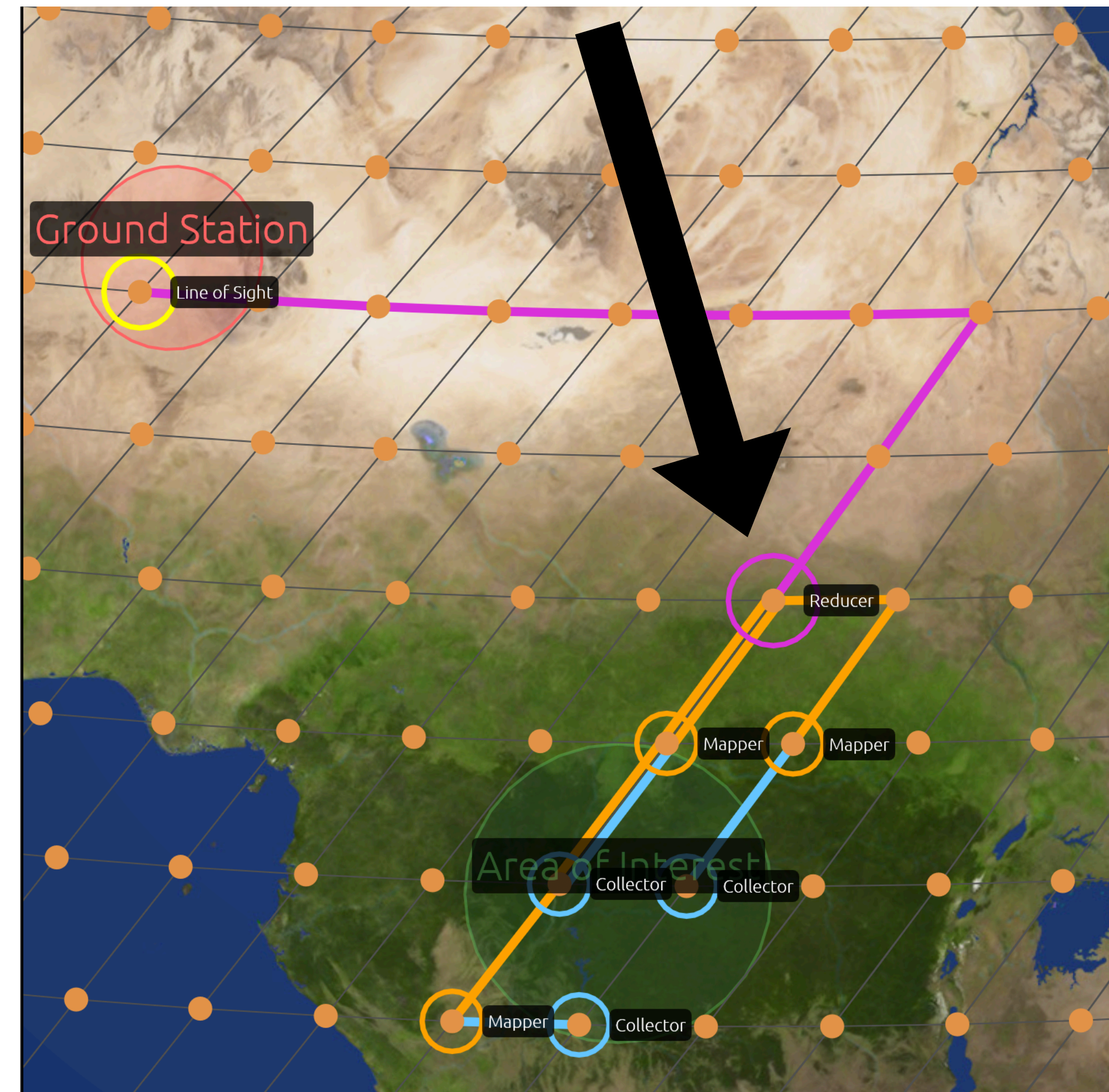


Optimisation 1 - Reducer placement

Closest to Line of sight (naïve)

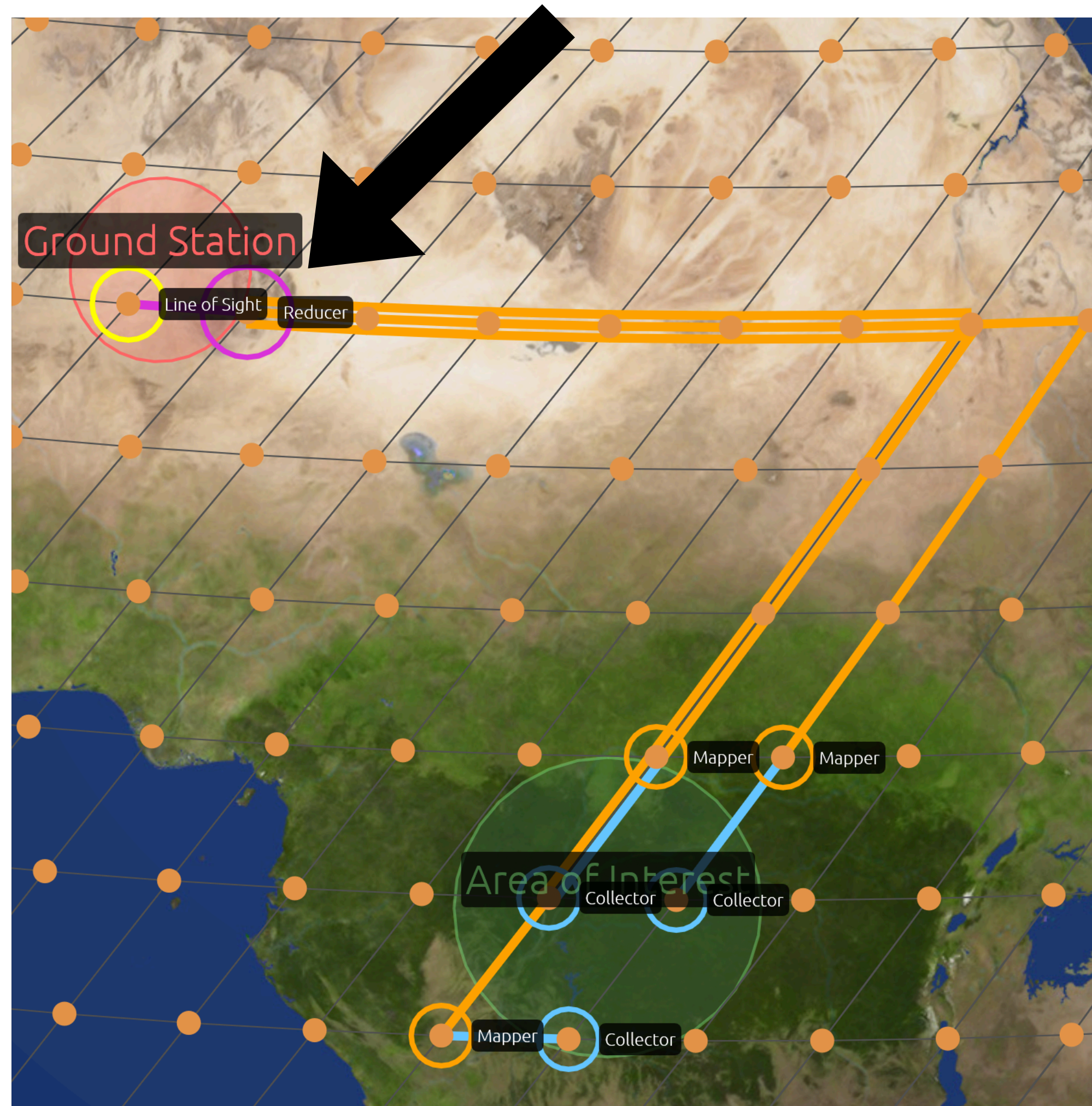


Closest to Mappers (optimisation)

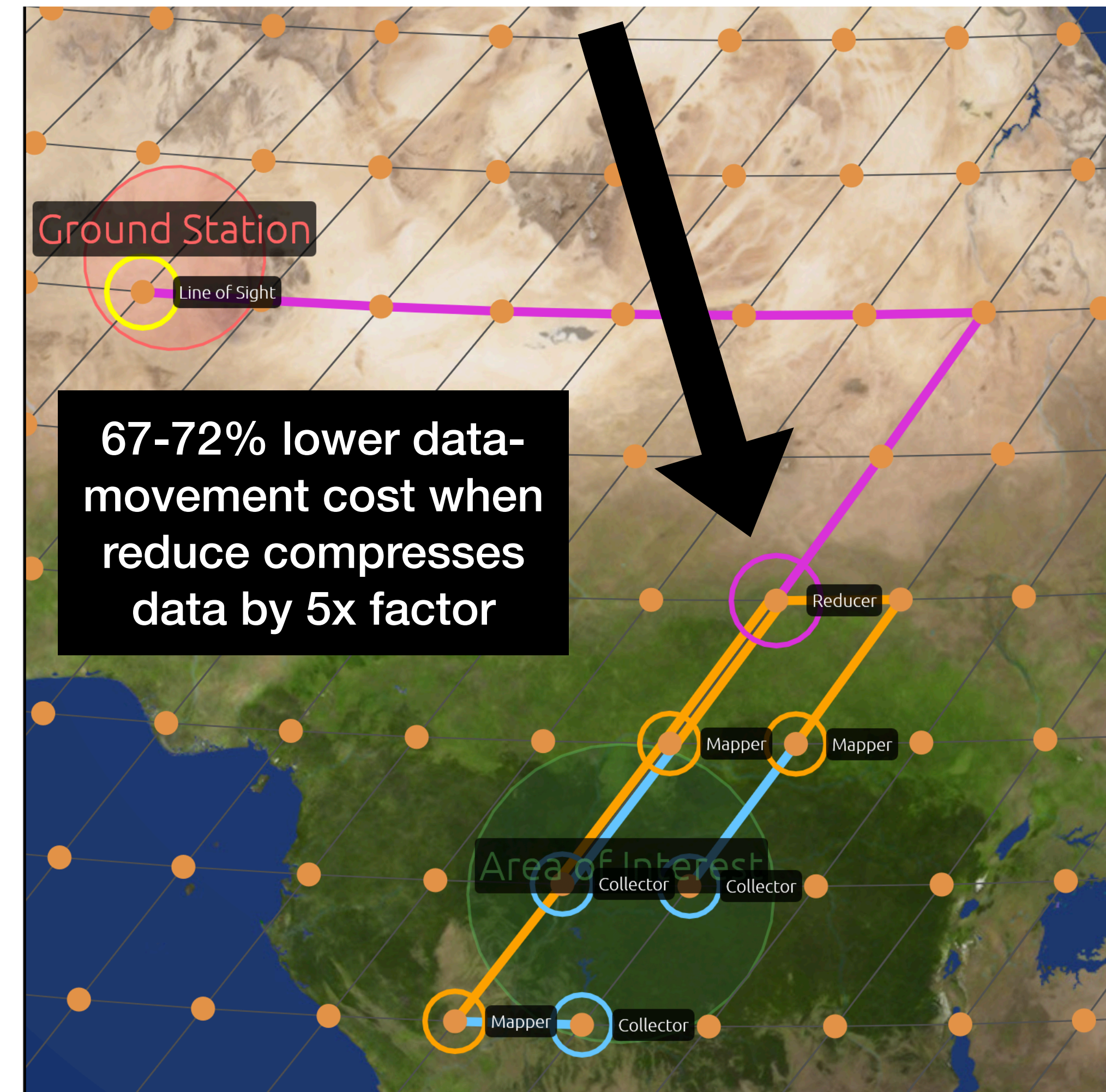


Optimisation 1 - Reducer placement

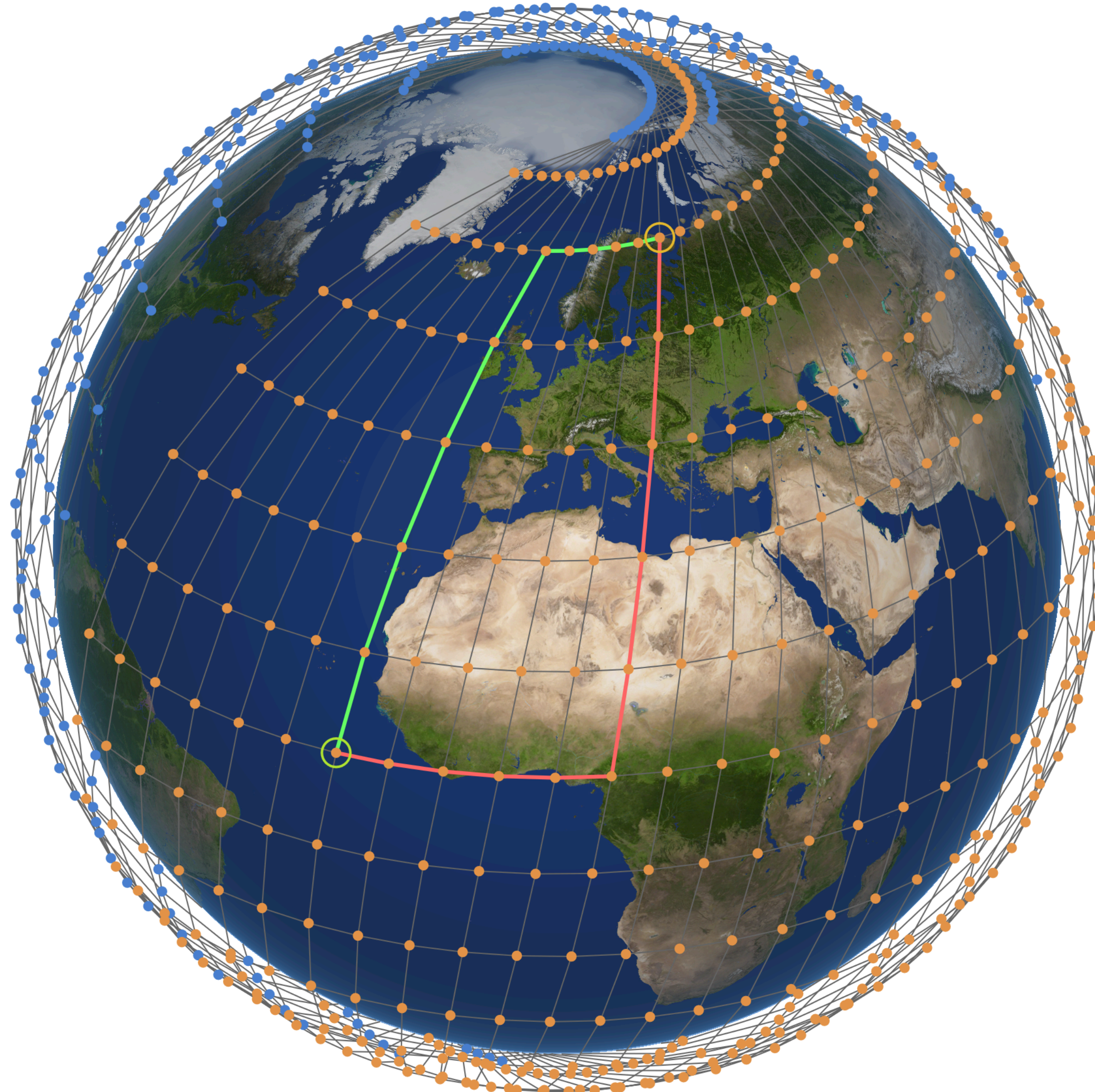
Closest to Line of sight (naïve)



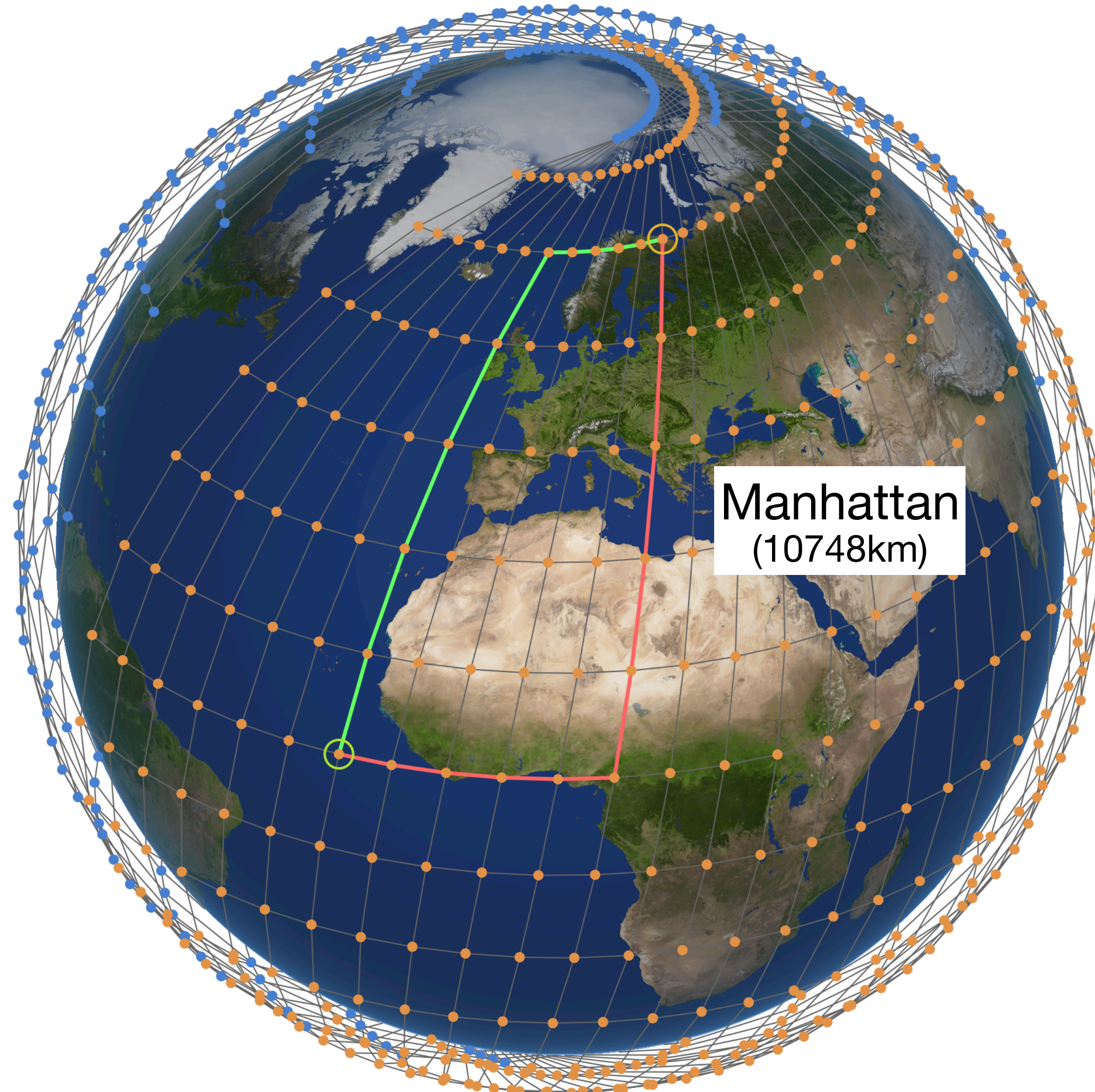
Closest to Mappers (optimisation)



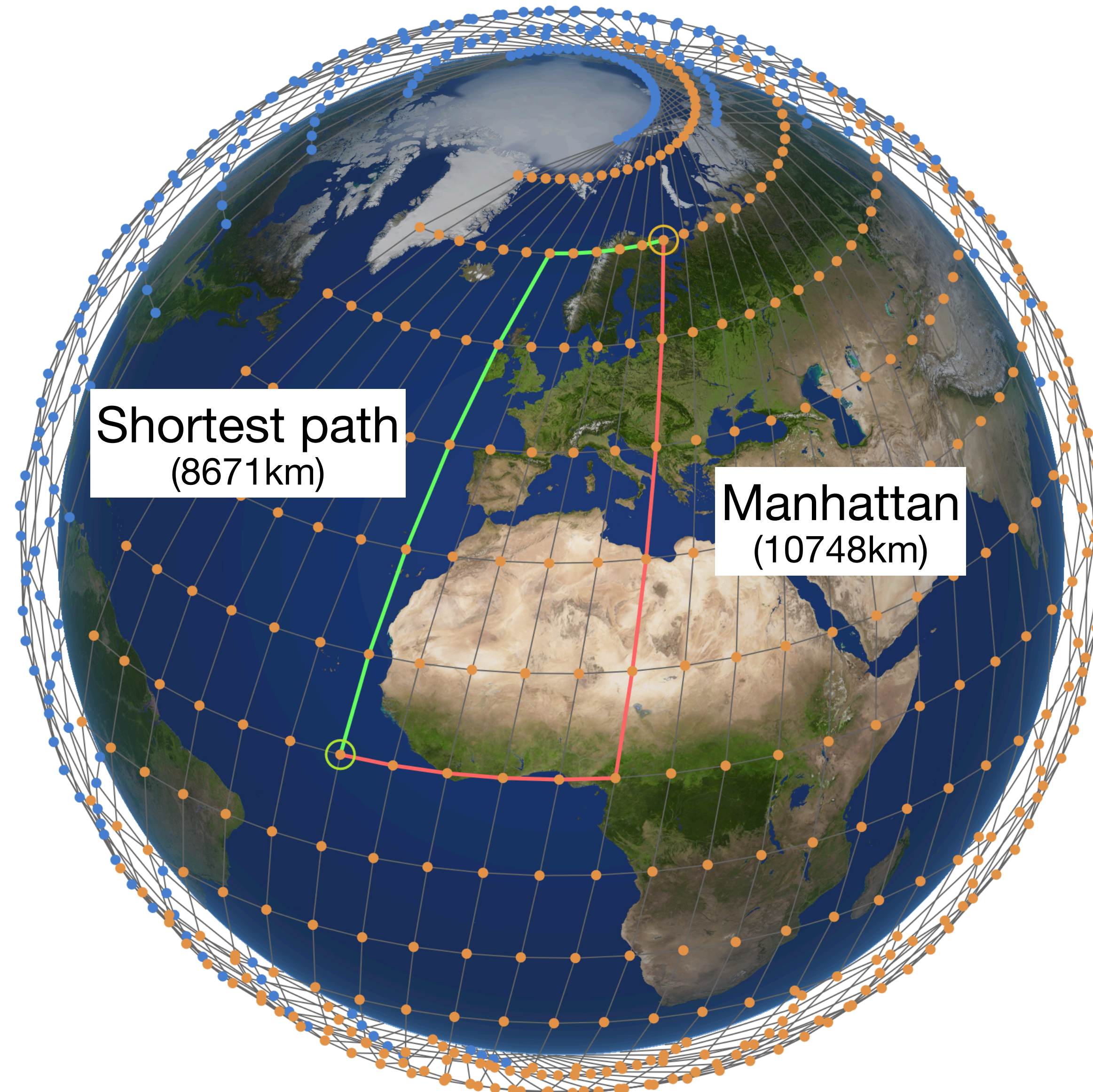
Optimisation 2 - Routing



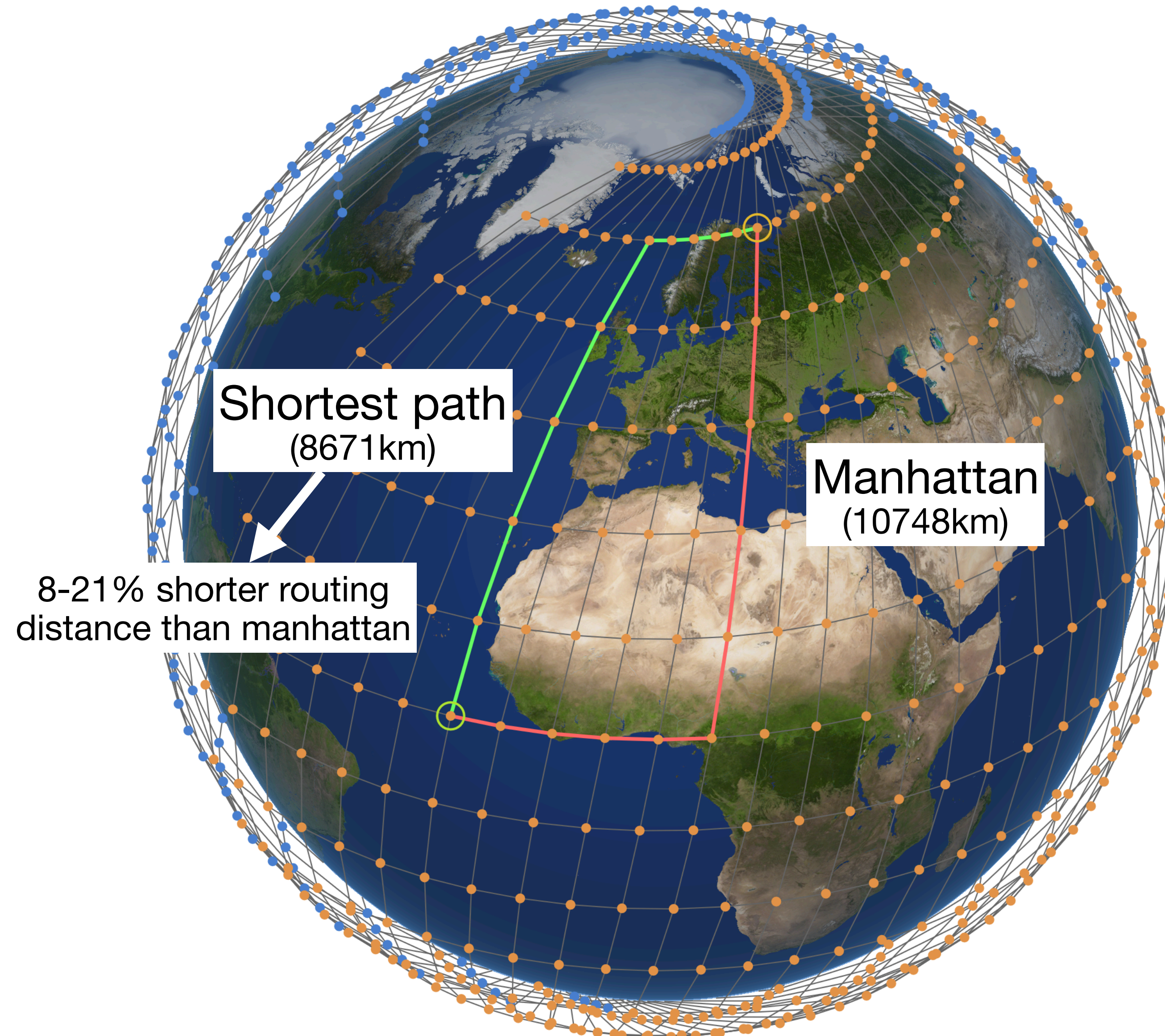
Optimisation 2 - Routing



Optimisation 2 - Routing



Optimisation 2 - Routing



Optimisation 3 - Bipartite Task Assignment

Optimisation 3 - Bipartite Task Assignment

Which **mapper** should process each **collector's** data?



Optimisation 3 - Bipartite Task Assignment

Which **mapper** should process each **collector's** data?



Cost matrix

cost = processing + #hops + transmission

	Mapper 1	Mapper 2	Mapper 3
Collector 1	1	2	9
Collector 2	1	8	9
Collector 3	9	2	3

Optimisation 3 - Bipartite Task Assignment

Which **mapper** should process each **collector's** data?



Cost matrix

cost = processing + #hops + transmission

	Mapper 1	Mapper 2	Mapper 3
Collector 1	1	2	9
Collector 2	1	8	9
Collector 3	9	2	3

Global optimal assignment
(Hungarian algorithm)

Optimisation 3 - Bipartite Task Assignment

Which **mapper** should process each **collector's** data?



Cost matrix

cost = processing + #hops + transmission

	Mapper 1	Mapper 2	Mapper 3
Collector 1	1	2	9
Collector 2	1	8	9
Collector 3	9	2	3

Global optimal assignment
(Hungarian algorithm)

61-79% lower global cost
vs random assignment

Implementation

Our Approach

Our Approach

SpaceCoMP Rust Library

LeoDOS (Leo Distributed OS)

Our Approach

SpaceCoMP Rust Library

LeoDOS (Leo Distributed OS)

Flight Software

Flight Software
Interface (Rust)

NASA cFS (C)

Core Flight Executive

Operating System
Abstraction Layer

Platform Support
Package

NOS3 Satellite Simulator

42

Buses,
Drivers

...

Our Approach

SpaceCoMP Rust Library

LeoDOS (Leo Distributed OS)

Flight Software

Flight Software
Interface (Rust)

NASA cFS (C)

Core Flight Executive

Operating System
Abstraction Layer

Platform Support
Package

NOS3 Satellite Simulator

42

Buses,
Drivers

...

Communication

CCSDS Protocols (Rust)

Application Layer
(Compression)

Transport Layer
(End-to-end messaging)

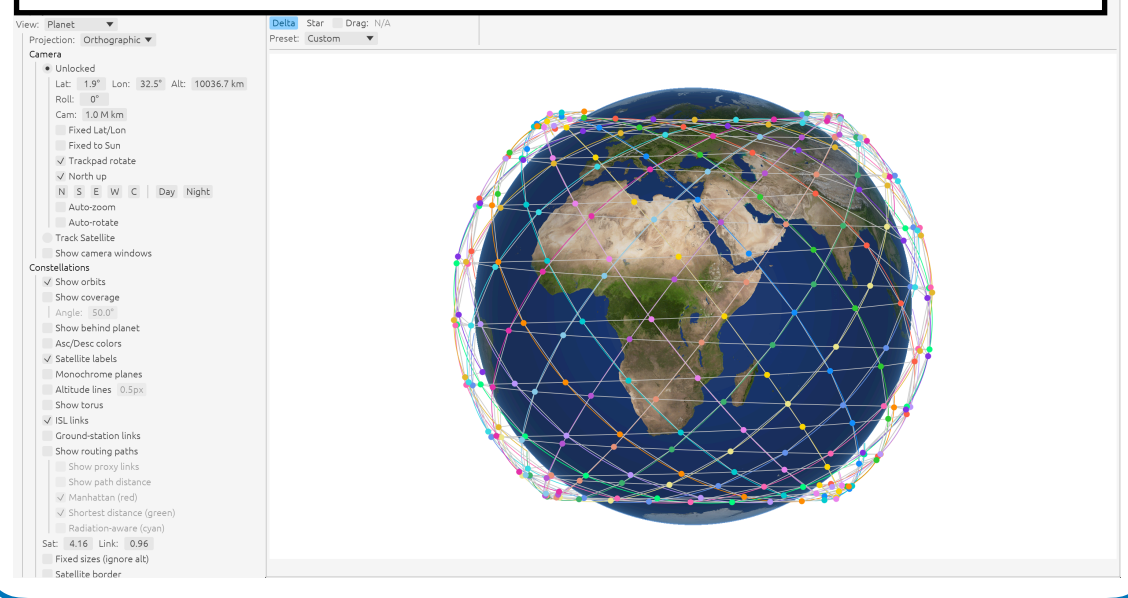
Network Layer
(Routing, Gossip)

Datalink Layer
(Packets, Framing, Security, ...)

Coding Layer
(Randomisation, Err. Correction, ...)

Physical Layer
(Modulation, Links)

Visualiser/Propagator (Rust)



Our Approach

SpaceCoMP Rust Library

LeoDOS (Leo Distributed OS)

Flight Software

Flight Software
Interface (Rust)

NASA cFS (C)

Core Flight Executive

Operating System
Abstraction Layer

Platform Support
Package

NOS3 Satellite Simulator

42

Buses,
Drivers

...

Communication

CCSDS Protocols (Rust)

Application Layer
(Compression)

Transport Layer
(End-to-end messaging)

Network Layer
(Routing, Gossip)

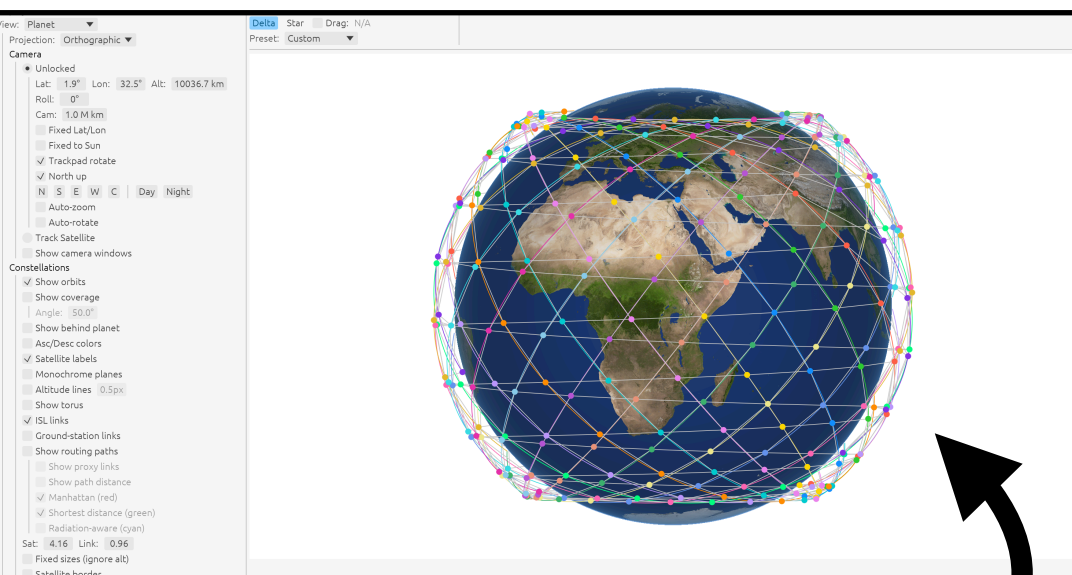
Datalink Layer
(Packets, Framing, Security, ...)

Coding Layer
(Randomisation, Err. Correction, ...)

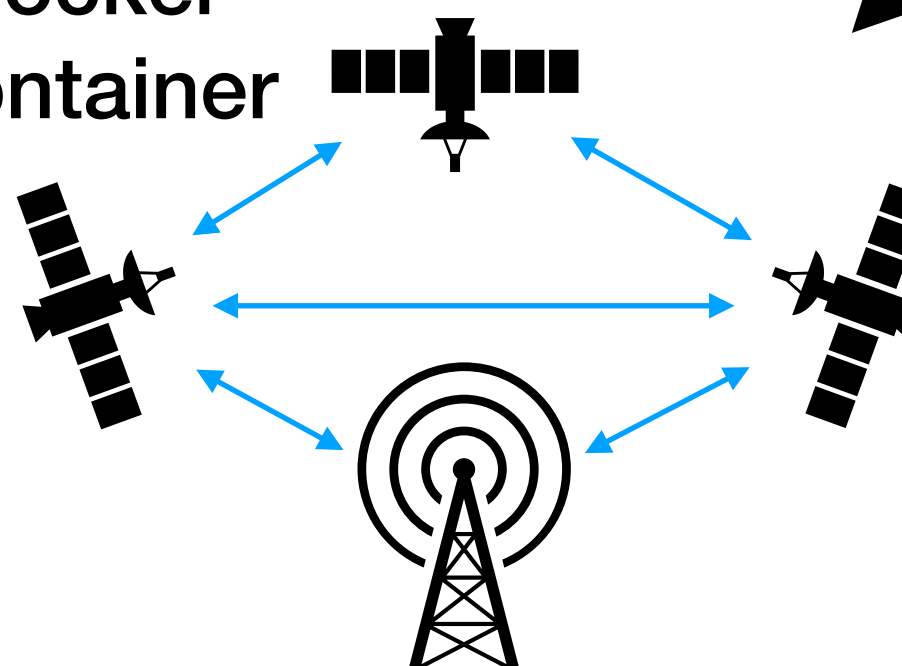
Physical Layer
(Modulation, Links)

Constellation Simulator

Visualiser/Propagator (Rust)



Docker
Container



Conclusions

Conclusions

SpaceCoMP aims to make the constellation a programmable computer for EO

- ➡ Process data where it is collected
- ➡ Coordinate work across satellites
- ➡ Downlink only the answers instead of raw imagery

Conclusions

SpaceCoMP aims to make the constellation a programmable computer for EO

- ➔ Process data where it is collected
- ➔ Coordinate work across satellites
- ➔ Downlink only the answers instead of raw imagery

Status:

- The presented work is a **first step** toward a working system
- Multiple **problems** remain to be solved
- **We are looking for input from the EO community**

Future work

Future work

Current Model

One job at a time

Future work

Current Model

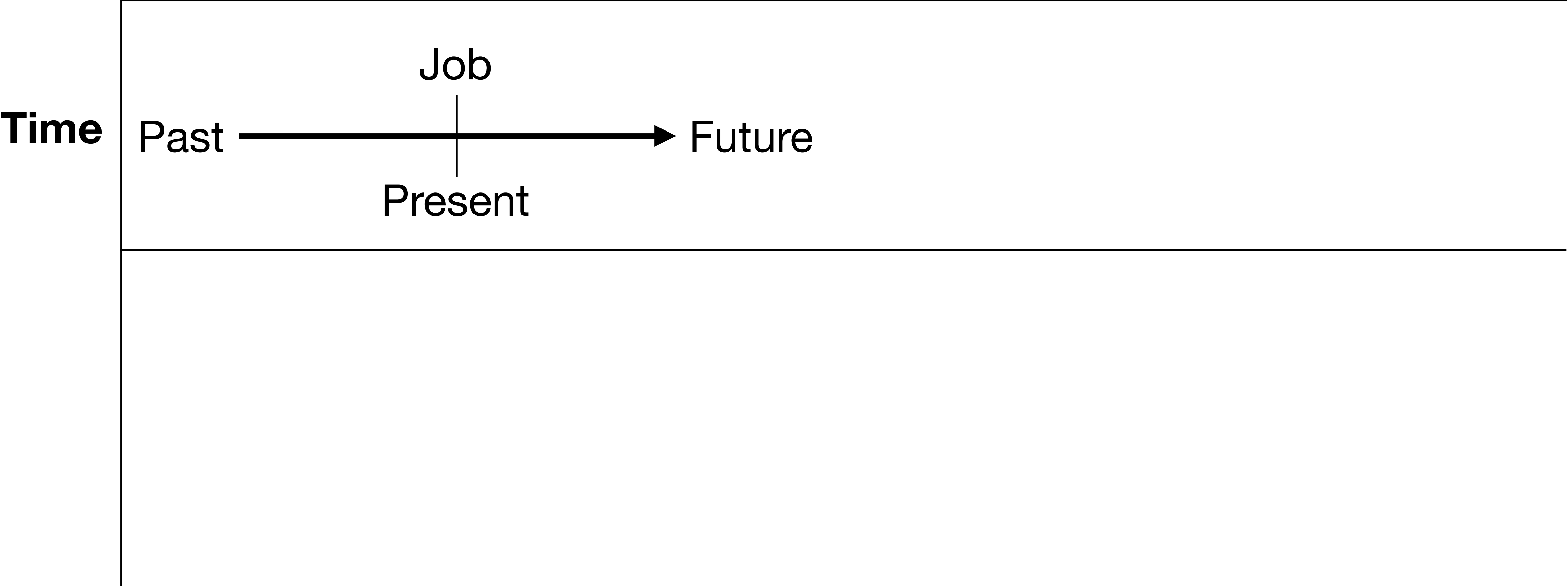
One job at a time



Future work

Current Model

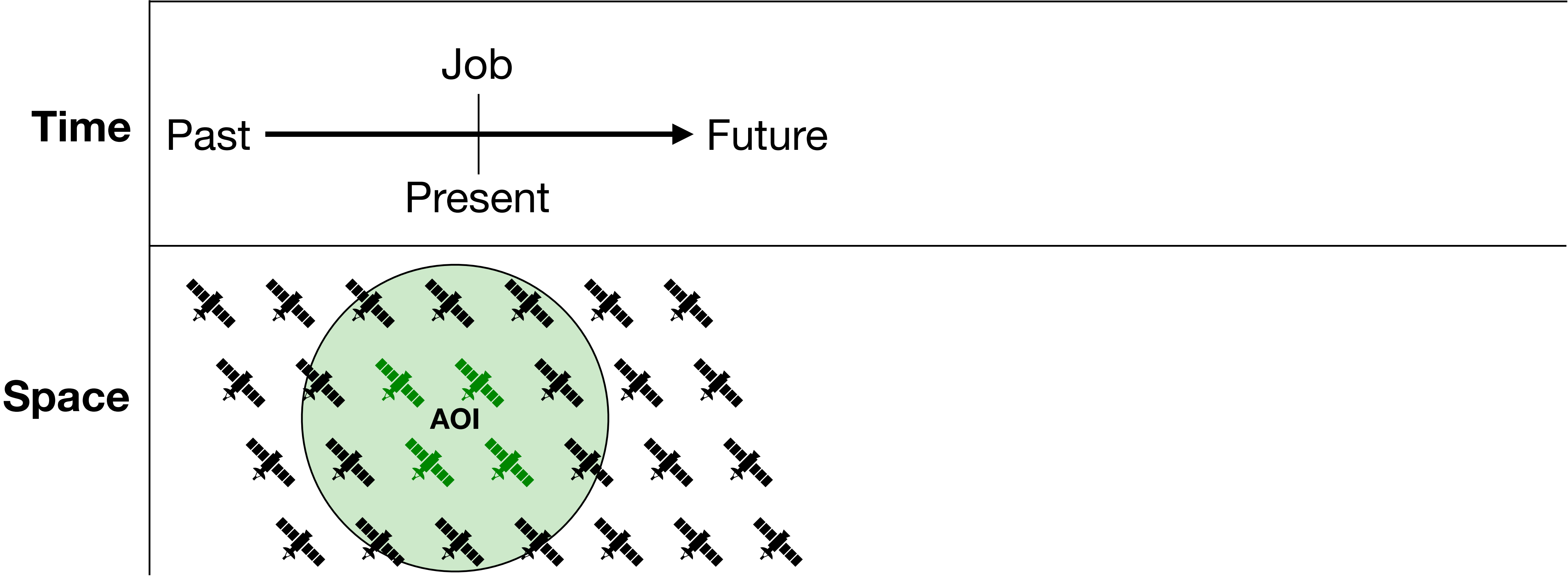
One job at a time



Future work

Current Model

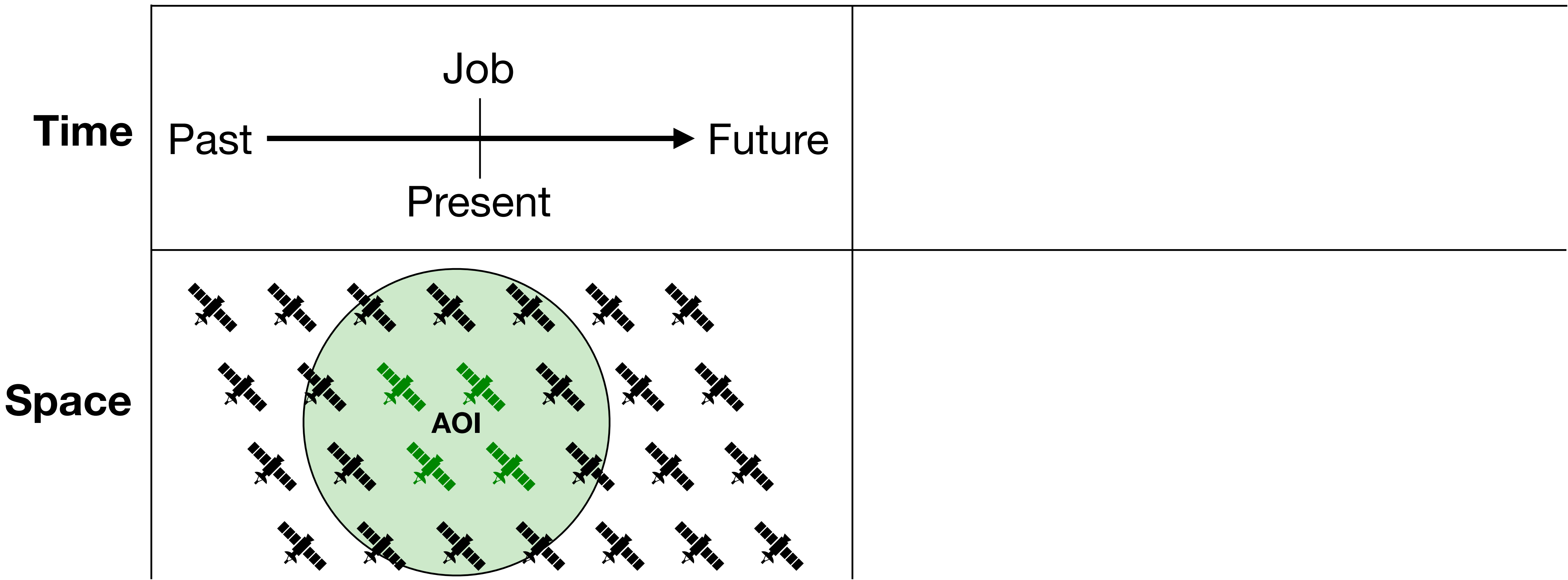
One job at a time



Future work

Current Model
One job at a time

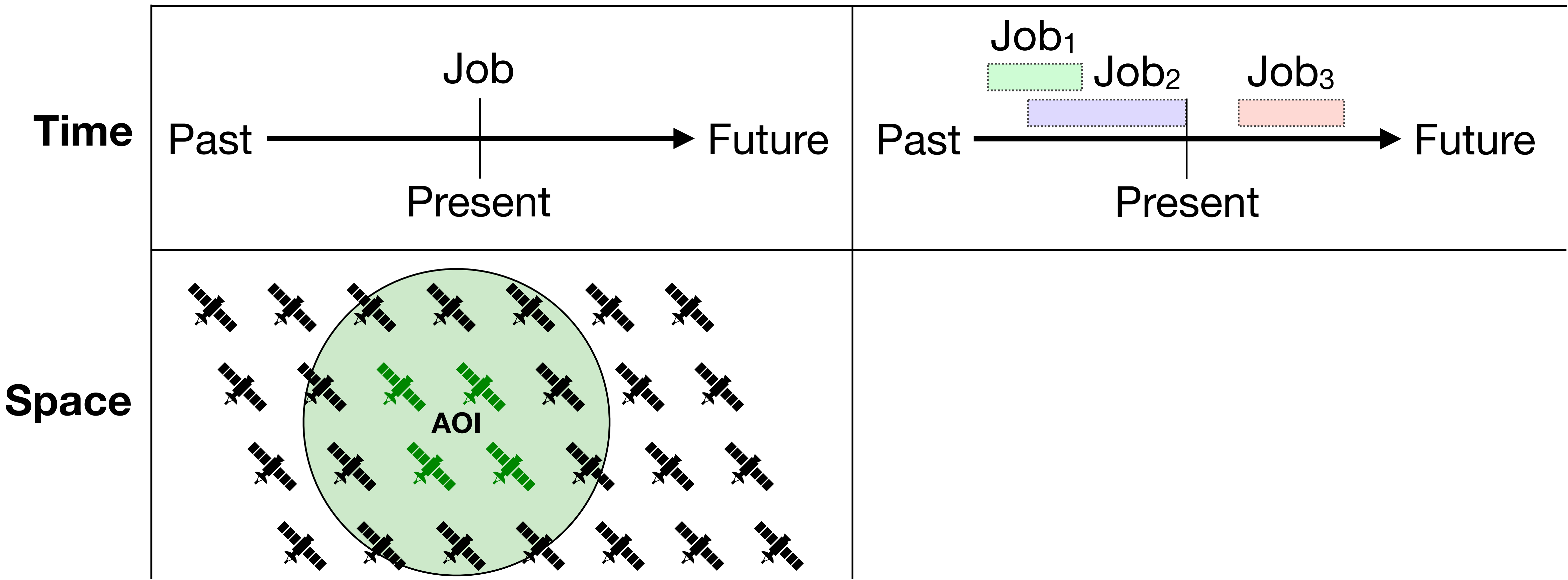
Future work
Multiple jobs, over time and space



Future work

Current Model
One job at a time

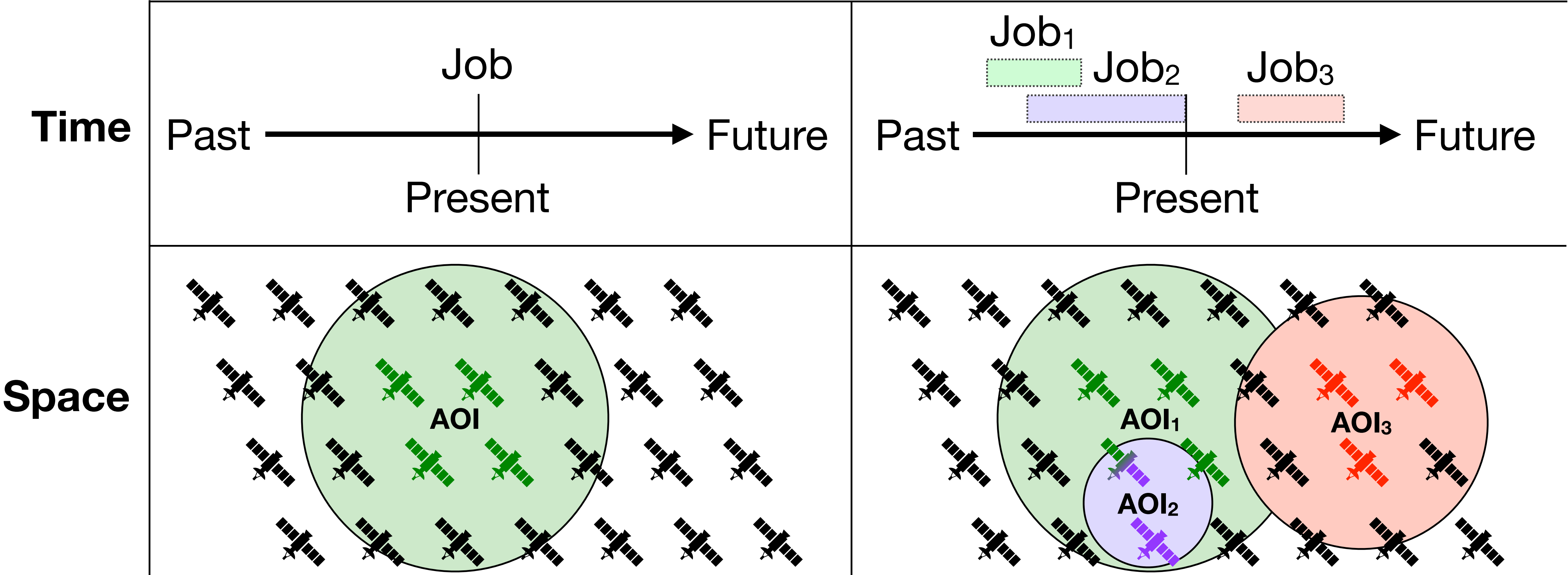
Future work
Multiple jobs, over time and space



Future work

Current Model
One job at a time

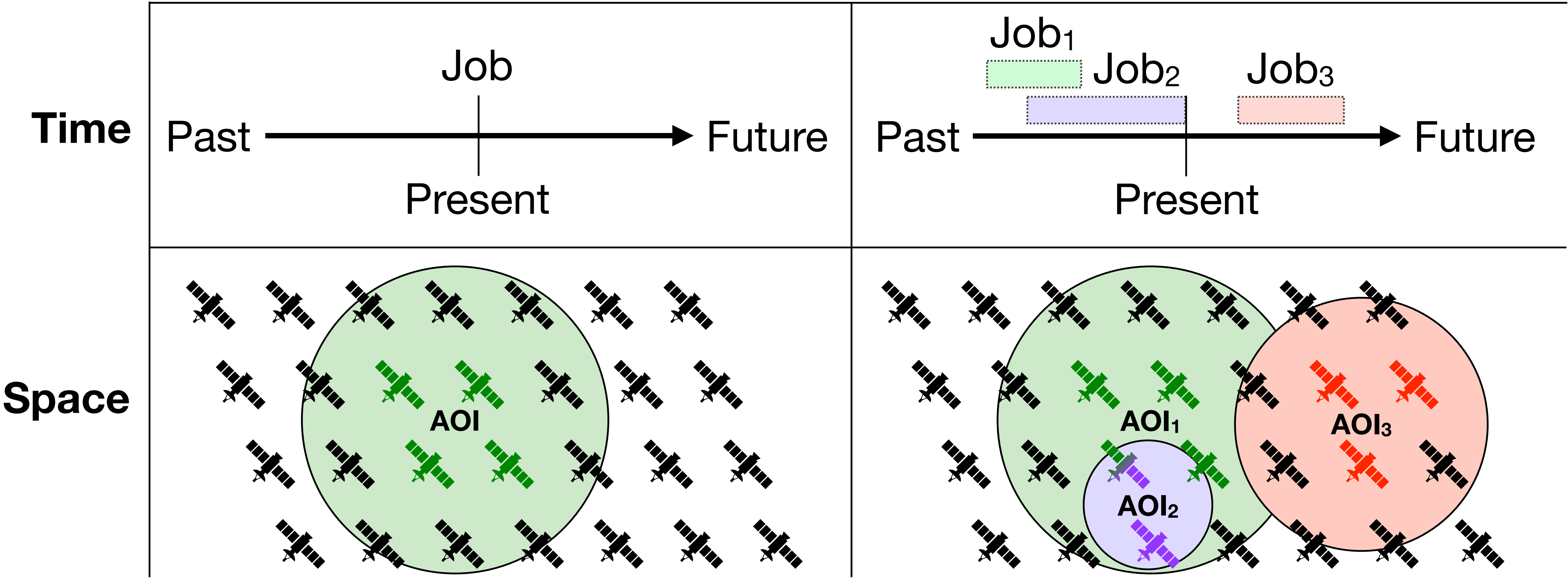
Future work
Multiple jobs, over time and space



Future work

Current Model
One job at a time

Future work
Multiple jobs, over time and space



Satellite Scheduling Problem





<https://leodos.org>

Lightspeed Data Compute for the Space Era

Thomas Sandholm*, Bernardo A. Huberman*, Klas Segeljakt*, and Paris Carbone*[†]

* *RISE - Research Institutes of Sweden*

thomas.sandholm@ri.se, bernardo.huberman@ri.se, klas.segeljakt@ri.se, paris.carbone@ri.se

[†] *KTH Royal Institute of Technology*

parisc@kth.se

Abstract—While thousands of satellites photograph Earth every day, most of that data never makes it to the ground because downlink bandwidth simply cannot keep up. Processing data in the Low Earth Orbit (LEO) zone offers promising capabilities to overcome this limitation. We propose *SpaceCoMP*, a MapReduce-inspired processing model for LEO satellite mesh networks. Ground stations submit queries over an area of interest; satellites collect sensor data, process it cooperatively at light-speed using inter-satellite laser links, and return only the results. Our compute model leverages space physics to accelerate computations on LEO megaconstellations. Our distance-aware routing protocol exploits orbital geometry. In addition, our bipartite match scheduling strategy places map and reduce tasks within orbital regions while minimizing aggregation costs. We have simulated constellations of 1,000–10,000 satellites showcasing 61–79% improvement in map placement efficiency over baselines, 18–28% over greedy allocation, and 67–72% reduction in aggregation cost. SpaceCoMP demonstrates that the orbital mesh is not merely useful as a communication relay, as seen today, but can provide the foundations for faster data processing above the skies.

Index Terms—Space Computing, MapReduce, Earth Observation

I. INTRODUCTION

Low-earth orbit (LEO) satellite constellations are generating data at rates that far exceed their capacity to transmit to the ground. A single high-resolution Earth observation satellite produces 1–2 TB of imagery per day [22], yet typical ground station contacts, lasting only 5–15 minutes per orbit, can transfer merely a fraction of this volume over bandwidth-constrained radio-frequency (RF) downlinks. With mega-constellations now reaching unprecedented scale, Starlink exceeded 7,500 operational satellites in 2024 with over 40,000 planned, and the EU’s IRIS² program commits €10.6B to European sovereign connectivity [5], [12], the aggregate data generation potential reaches petabytes per day. Yet global ground station infrastructure remains the bottleneck: RF spectrum is shared and regulated, weather causes signal attenuation, and ground stations must be within line-of-sight during brief orbital passes. This *downlink wall* fundamentally limits the value that can be extracted from space-based sensing.

The emergence of optical inter-satellite links (ISL) offers a transformative opportunity. Unlike RF links that must traverse Earth’s atmosphere, optical ISLs operate in vacuum at the speed of light, approximately 47% faster than signals in terrestrial fiber [7], with bandwidth exceeding 10 Gbps per

TABLE I RF GROUND LINKS VS. OPTICAL INTER-SATELLITE LINKS (ISLs)		
Property	RF Ground Links	Optical ISLs
Medium	Atmosphere (lossy)	Vacuum
Propagation speed	≈200,000 km/s	299,792 km/s (<i>c</i>)
Bandwidth	Shared spectrum, <1 Gbps	10–100+ Gbps/link
Availability	Weather-dependent	Always-on
Ground infrastructure	Required	Not required
Long-haul latency	High (via ground hops)	Low (direct path)
Atmospheric loss	Yes (rain, clouds)	None

link [16]. Table I summarizes the key differences. Modern LEO satellites equipped with laser terminals form mesh networks where any satellite can route data to any other through a sequence of hops, creating a distributed infrastructure with high-bandwidth internal connectivity but constrained external bandwidth to ground. From a computational lense the LEO architecture features thousands of interconnected compute nodes and data is naturally partitioned across collectors. The cost of performing external I/O essentially mirrors similar conditions that motivated MapReduce in terrestrial data centers two decades ago [4]. A key insight is that *processing data in orbit before transmission can dramatically reduce the potential of downlink saturation*, converting terabytes of raw sensor data into megabytes of actionable results.

However, distributed computing in LEO presents novel challenges absent from terrestrial systems. Satellites orbit at 7.5 km/s, completing a full orbit in approximately 95 minutes. The network topology changes continuously: inter-plane link distances vary by up to 40% over each orbital period as planes converge near the poles and diverge at the equator. Near polar regions, satellites from adjacent planes pass each other at relative velocities exceeding 15 km/s, forcing temporary link disconnection. Task placement must therefore account not just for hop count but for actual transmission distance and time-varying link geometry. Furthermore, satellites remain in proximity to a ground station for only 5–15 minutes before handover [19], requiring coordination of data collection, processing, and result delivery within narrow windows. Existing cloud software, designed for static datacenter topologies,

arXiv paper

<https://leodos.org>

arXiv:2601.17589v2 [cs.DC] 2 Feb 2026

Lightspeed Data Compute for the Space Era

Thomas Sandholm*, Bernardo A. Huberman*, Klas Segeljakt*, and Paris Carbone*†

* *RISE - Research Institutes of Sweden*

thomas.sandholm@ri.se, bernardo.huberman@ri.se, klas.segeljakt@ri.se, paris.carbone@ri.se

† *KTH Royal Institute of Technology*

parisc@kth.se

Abstract—While thousands of satellites photograph Earth every day, most of that data never makes it to the ground because downlink bandwidth simply cannot keep up. Processing data in the Low Earth Orbit (LEO) zone offers promising capabilities to overcome this limitation. We propose *SpaceCoMP*, a MapReduce-inspired processing model for LEO satellite mesh networks. Ground stations submit queries over an area of interest; satellites collect sensor data, process it cooperatively at light-speed using inter-satellite laser links, and return only the results. Our compute model leverages space physics to accelerate computations on LEO megaconstellations. Our distance-aware routing protocol exploits orbital geometry. In addition, our bipartite match scheduling strategy places map and reduce tasks within orbital regions while minimizing aggregation costs. We have simulated constellations of 1,000–10,000 satellites showcasing 61–79% improvement in map placement efficiency over baselines, 18–28% over greedy allocation, and 67–72% reduction in aggregation cost. SpaceCoMP demonstrates that the orbital mesh is not merely useful as a communication relay, as seen today, but can provide the foundations for faster data processing above the skies.

Index Terms—Space Computing, MapReduce, Earth Observation

I. INTRODUCTION

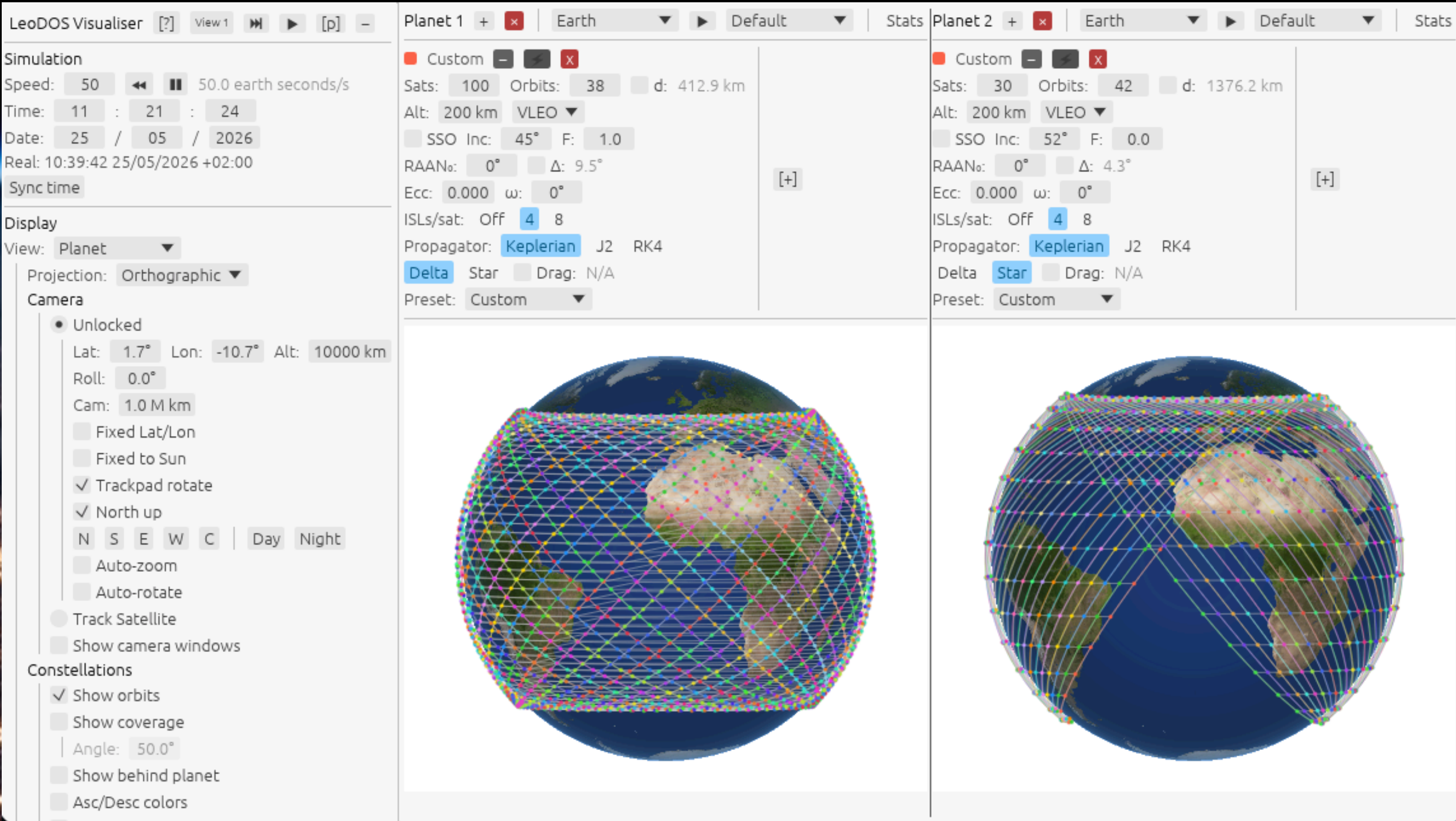
Low-earth orbit (LEO) satellite constellations are generating data at rates that far exceed their capacity to transmit to the ground. A single high-resolution Earth observation satellite produces 1–2 TB of imagery per day [22], yet typical ground station contacts, lasting only 5–15 minutes per orbit, can transfer merely a fraction of this volume over bandwidth-constrained radio-frequency (RF) downlinks. With mega-constellations now reaching unprecedented scale, Starlink exceeded 7,500 operational satellites in 2024 with over 40,000 planned, and the EU’s IRIS² program commits €10.6B to European sovereign connectivity [5], [12], the aggregate data generation potential reaches petabytes per day. Yet global ground station infrastructure remains the bottleneck: RF spectrum is shared and regulated, weather causes signal attenuation, and ground stations must be within line-of-sight during brief orbital passes. This *downlink wall* fundamentally limits the value that can be extracted from space-based sensing.

The emergence of optical inter-satellite links (ISL) offers a transformative opportunity. Unlike RF links that must traverse Earth’s atmosphere, optical ISLs operate in vacuum at the speed of light, approximately 47% faster than signals in terrestrial fiber [7], with bandwidth exceeding 10 Gbps per

TABLE I RF GROUND LINKS VS. OPTICAL INTER-SATELLITE LINKS (ISLs)		
Property	RF Ground Links	Optical ISLs
Medium	Atmosphere (lossy)	Vacuum
Propagation speed	≈200,000 km/s	299,792 km/s (<i>c</i>)
Bandwidth	Shared spectrum, <1 Gbps	10–100+ Gbps/link
Availability	Weather-dependent	Always-on
Ground infrastructure	Required	Not required
Long-haul latency	High (via ground hops)	Low (direct path)
Atmospheric loss	Yes (rain, clouds)	None

link [16]. Table I summarizes the key differences. Modern LEO satellites equipped with laser terminals form mesh networks where any satellite can route data to any other through a sequence of hops, creating a distributed infrastructure with high-bandwidth internal connectivity but constrained external bandwidth to ground. From a computational lense the LEO architecture features thousands of interconnected compute nodes and data is naturally partitioned across collectors. The cost of performing external I/O essentially mirrors similar conditions that motivated MapReduce in terrestrial data centers two decades ago [4]. A key insight is that *processing data in orbit before transmission can dramatically reduce the potential of downlink saturation*, converting terabytes of raw sensor data into megabytes of actionable results.

However, distributed computing in LEO presents novel challenges absent from terrestrial systems. Satellites orbit at 7.5 km/s, completing a full orbit in approximately 95 minutes. The network topology changes continuously: inter-plane link distances vary by up to 40% over each orbital period as planes converge near the poles and diverge at the equator. Near polar regions, satellites from adjacent planes pass each other at relative velocities exceeding 15 km/s, forcing temporary link disconnection. Task placement must therefore account not just for hop count but for actual transmission distance and time-varying link geometry. Furthermore, satellites remain in proximity to a ground station for only 5–15 minutes before handover [19], requiring coordination of data collection, processing, and result delivery within narrow windows. Existing cloud software, designed for static datacenter topologies,



<https://leodos.org>

arXiv paper