KP LABS
Mission Complete

IBM **Research**
Europe

SYSTEM ANALYSIS REPORT

SPACE-BASED DATA CENTRES PROJECT

AUTHORS: Jonas Weiss[1], Patricia Sagmeister[1], Martin Schmatz[1], Dinesh Verma[1], Sagar Tayal[2], Ingmar Meijr[1], Thomas Morf[1], Dawid Lazaj[3], Agata Wijata[3], Jakub Nalepa[3],

AFFILIATIONS: 1) IBM Research, 2) IBM Corporation, 3) KP Labs

DATE: 2024-03-19

VERSION: 2.0

Review History

Version	Date	Changes	Editor
0.1	Jun 26, 2023	Document Creation	Jonas Weiss
0.2	July 3, 2023	Added Section 7: Software Stack	Martin Schmatz
0.21	July 6, 2023	Quantitative model to section	Jonas Weiss
0.22	August 1, 2023	Added Network Architecture	Dinesh Verma
0.3	August 14	Nomenclature, Network review, added structure for simulator	Jonas Weiss

2023

0.4	August 24	Orbit architecture & Network Model	Jonas Weiss
0.5	October 12	Added SDC Hardware Section, included revised Network Architecture	Jonas Weiss Sagar Tayal Dinesh Verma
0.6	October 25	Security Section added	Patricia Sagmeister
1.0	March 15, 2024	Clean-up and complement all sections for submission	Jonas Weiss

TABLE OF CONTENTS

1. ACRONYMS

Acronym	Explanation
AI	Artificial Intelligence
ARGP	Argument of Periapsis
CNSA	Commercial National Security Algorithms
COTS	Commercial off the Shelf (applies e.g. to electronic components)
CSR	Certified Signature Request
DC	Data Center
DCAP	Data Center Attestation Primitives
DDR	Double Data Rate (Memory)
DL	Deep Learning (in the context of artificial intelligence)
ECC	(Orbital) Eccentricity
EHF	Extremely High Frequency
ESA	European Space Agency
FHM	Fully Homomorphic (Encryption)
FM	Foundation Model (in the context of artificial intelligence)
FOM	Figure of Merit
G(Fl)Ops	Giga (Floating Point) Operations per second
GB	Giga Byte

Gbps	Giga bit per seconds (in the context of data rate and bandwidth)
GEO	Geostationary Orbit
GPS	Global Positioning System
HPC	High Performance Computing
HSM	Hardware Security Module
HW	Hardware
IAM	Identity Access Management
IBM	International Business Machines (Corporation)
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
LEO	Lower Earth Orbit
Mbps	Megabits per second
MS	Microsoft
mTLS	Mutual TLS
NIST	National Institute of Standards
OCT	Optical Communication Terminal
OKD	(OpenShift) Origin Community Distribution
Ops	Operations per second
OS	Operating System

PKI	Public Key Infrastructure
QSC	Quantum Safe Cryptography
RAAN	Right Ascension of the Ascending Node
RD	Reference Document
SA	System Architecture
SDA	Space Development Agency
SDC	Space (based) Data Center
SHF	Super High Frequency
SOC	System On a Chip
SW	Software
T(Fl)Ops	Tera (Floating Point) Operations per second
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TOC	Table of Content
TOS	Top of the Rack Switch
TPM	Trusted Platform Module
UI	User Interface
VBA	Visual Basic Script
VM	Virtual Machine

VPN	Virtual Private Network
WP	Work Package

2. REFERENCE DOCUMENTS

ID	Document
RD-1	User-Level Requirements Report
RD-2	Use-Cases Report
RD-3	System Level Specification
RD-4	This document

3. INTRODUCTION

Authored by: Jonas Weiss

3.1. BACKGROUND

This document is part of the study on **Space-based Data Centres** (SDCs), with requirement details available in the technical tender document 7732.

This document aggregates the output of the technical work packages WP-4, WP-5, WP-6, WP-7, and benchmarking results from WP-9 of the ESA Space-based Data Centres Study. With a single document instead of individuals, we believe that the overall story and style can be better preserved, and it will be easier for authors and readers to find relevant information. The main purpose is to provide sufficient information to build and understand the system simulator of section 9.

3.2. OVERVIEW & WORKPACKAGES

This document contains the consolidated results from below work packages. No other/separate reports will be compiled for these work packages.

Table 1 Work package Breakdown

ID	Title	Responsible Contractor
WP-1	Use-Cases and Added Value	KP Labs
WP-2	User-Level Requirements	KP Labs
WP-3	System-Level Metrics & Requirements	KP Labs
WP-4	System-Level Architecture	IBM
WP-5	Workload Analysis & Software Stack	IBM
WP-6	Network Architecture & Management	IBM
WP-7	Payload/DC Architecture	IBM
WP-8	Enabling Technologies and Roadmaps	KP Labs
WP-9	System Simulator	IBM
WP-10	Commercialization & Business Model	IBM
WP-11	Industry Engagement & Social Media	KP Labs
WP-12	Project Management	IBM

3.3. NOMENCLATURE

3.3.1. SPACE DATA CENTER

Figure 1 depicts the overall concept of a distributed space data center (SDC), where we also refer to a single satellite of the network as "a space data center". Data collected from clients is shared with a network of connected SDC satellites, which provide compute, network and downlink capacities to the clients, in many ways similar to terrestrial public cloud data centers.

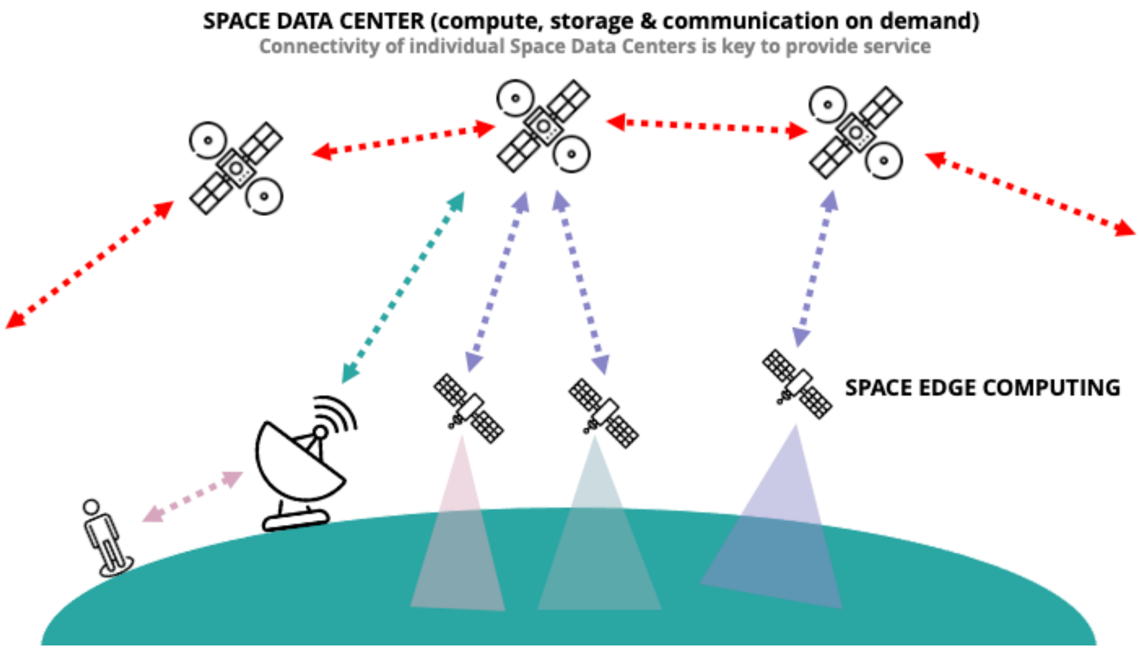


Figure 1 Network of Space Data Centers and client satellites with limited edge processing capabilities, which are complemented by the Space Data Center Network

3.3.2. SATELLITE CONSTELLATION

A satellite constellation is a group of artificial satellites working together as a system. Unlike a single satellite, a constellation can provide permanent global or near-global coverage, such that at any time everywhere on Earth at least one satellite is visible. Satellites are typically placed in sets of complementary orbital planes and connect to globally distributed ground stations. They may also use inter-satellite communication [1] .

In the context of Space Data Centers, we consider the collective of all connected SDCs as a constellation, which provides in-space on-demand compute, storage/buffering and communication services through a unified access/interface scheme and many redundant access points.

For future scenarios, we may as well extend the notion of the constellation towards an "ad hoc constellation", i.e., a constellation that consists of dynamically on- and off-boarded satellites into a constellation of SDCs. This will be based on availability and need of specific satellites to contribute to specific tasks, and the agreement of their respective owners/operators to (temporarily) become part of the constellation. Clearly, such an extension calls for transparent and open standards and protocols.

3.3.3. SDC (NODE) SATELLITE

In this document, we refer to an **SDC node** satellite as a spacecraft which connects to the SDC constellation for the purpose of sharing its resources and services to complement and extend the constellation's technical and service capabilities. While the term "node" is mostly used in the context of networking, we also use "**SDC pod**" (see e.g., Figure 10), to describe the compute payload of a single SDC satellite, but mostly in the context of compute hardware (hierarchies).

We expect SDC nodes to mainly be dedicated spacecrafts making available significant compute, storage, and communication bandwidth to the constellation, but in certain scenarios also satellites with at least one of compute, storage or communication capabilities may be considered an SDC node.

There are SDC concepts where the data center itself is considered a massive aggregation of many, individually launched satellites that are mechanically connected to form a massive in-space infrastructure. In this document and study, we exclude this concept from our considerations to be able to focus on a distributed version thereof, which provides many more access points to client satellites and in our view better meets space specific operational constraints.

3.3.4. CLIENT SATELLITE

In this document, we refer to a **client satellite** as a spacecraft which connects to the SDC constellation for the purpose of accessing its resources and services to complement and extend its own technical and service capabilities.

We usually think of client satellites as smaller, strongly resource constraint satellites. Never-the-less, for specific scenarios, also larger spacecraft may profit from, and thus connect to an SDC constellation. In the extreme, a client satellite could even be replaced with an entire constellation, which for a specific task could benefit from additional on-demand or shared resources from an SDC.

3.3.5. GROUND STATION

A ground station, Earth station, or Earth terminal is a terrestrial station designed for extraplanetary telecommunication with spacecraft (constituting part of the ground segment of the spacecraft system). Ground stations communicate with SDCs by transmitting and receiving radio waves in the super high frequency (SHF), extremely high frequency (EHF) bands (e.g., microwaves) or by using optical signals, i.e. laser communication. When a ground station successfully transmits signals to a spacecraft (or vice versa), it establishes a telecommunications link. A principal telecommunications device of the ground station is the parabolic antenna or optical mirror.

When a spacecraft or SDC is within a ground station's line of sight, the station is said to have a view of the spacecraft (see pass). A spacecraft can communicate with more than one ground station at a time and ground stations may communicate with multiple satellites at the time. [2]

3.3.6. DOWNSTREAM TASK

A downstream task in the context of SDCs is a task that relates to end-user-requirements, usually on the ground. Examples are e.g., detecting and reporting on wildfire location, intensity, or the availability of access and escape routes. Downstream tasks may require the use of a single satellite, but for many cases may also benefit from having data available from several sources and having them combined in more sophisticated models. The term downstream task here is loosely related to the same term used in machine learning and AI, where it relates to the (final) task with dependencies on previous (sub)tasks in the context of transfer learning.

3.4. QUANTIFIABLE OUTPUT FOR SIMULATION TOOL

While substantial work was invested in reviewing state-of-the-art and other related technologies, the main deliverable of this study is a simulation and benchmarking tool that facilitates comparing different system arrangements for specific applications scenarios. For the latter, predefined figures of merits (FOMs) will be compared as a function of design-decisions at different levels.

To implement such a simulation or assessment tool, the output of every work package is, besides a concise description of the findings and interdependencies, a contribution to the simulation work package in the form of quantitative relations between system design choices and implementations.

As an example: The system architecture foresees 4 SDC nodes with all-to-all connections (if visibility permits). This yields 3 ultra-high bandwidth optical intersatellite terminals per unit. Furthermore, every node shall be able to serve 10 clients simultaneously with 100 Mbit/s minimum bandwidth each. The latter yields 1 Gigabit overall upstream capacity per node, with 360 x 90-degree visibility, by assuming a lower orbit of the clients than the SDC nodes. Whether this is implemented with multiple smaller optical terminals (likely) or a single "super-terminal" (unlikely), is subject to the node-design, and not the architecture, but the overall data-link capacity required is a function of the architectural choice and thus a quantifiable output to the simulation model.

This example thus yields the following equations, which can be implemented in the benchmark simulation tool and help estimate power consumption, weight, and cost of a single SDC node/Pod.

Design choices:
- SDC connected to 4 other SDCs within constellation ($\#Connections_{High\_BW}$)
- Every SDC to serve 10 client nodes concurrently ($\#Clients_{per\_node}$)
- 1 terminal is required per client served ($\#Clients_{per\_terminal}$)

As simple as below equations may seem, they are required to estimate hardware requirements which in turn allow to estimate benchmark figures of merits (FOMs).

$$\#Terminals_{high\_BW} = \#Connections_{high\_BW} \quad \text{Eq. 1}$$

$$\#Terminals_{low\_BW} = \#Clients_{per\_node} / \#Clients_{per\_terminal} \quad \text{Eq. 2}$$

Figure 1 depicts how different chapters and study components contribute to each other and finally provide the necessary quantifiable input to facilitate building a self-contained system simulator.

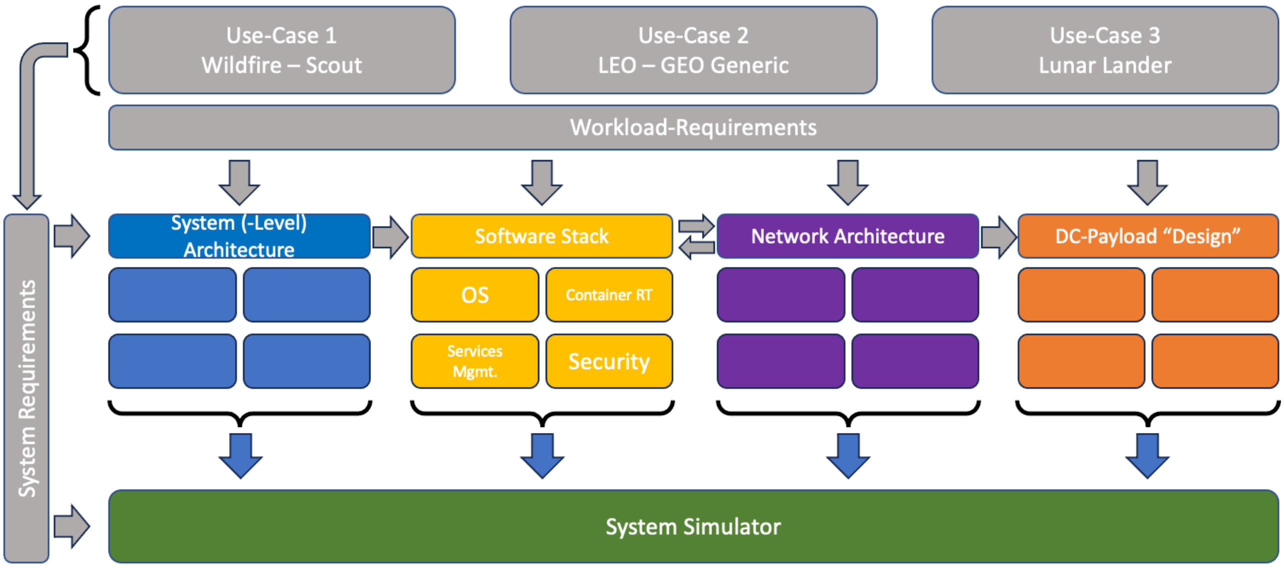


Figure 2 System Simulator Dependency from Use-Cases and Work Packages

4. SYSTEM-LEVEL REQUIREMENTS SUMMARY

Authored by: Jonas Weiss

A comprehensive list of requirements for SDCs has been derived in RD-3, for the three sample-use cases: Mothership-Scout (Wildfires), LEO-GEO Observation, Lunar-Lander. To support key architectural decisions, a condensed requirements list has been derived from RD-3 and validated against constraints of different architectures.

4.1. USE CASE 1 – MOTHERSHIP & SCOUT (WILDFIRES)

Table 2 Key Requirements for Wildfire Use Case 1

#	REQUIREMENT	MINIMUM VALUE
SR-U1-DP-6	Mothership processing capability (float32)	8 TFLOPs
SR-U1-DP-7	Mothership cold storage (5 days worth of raw data, 10 bands, 5000x5000 pixel, 8 bit, every 30 seconds)	4 Tbyte
SR-U1-LB-1	Mothership to Scout Telemetry bandwidth	30 kbps
SR-U1-LB-x	Scout to Mothership Data-Link bandwidth (1200 x 1200 x 3 bands every 30 seconds + overhead)	200 kbps

4.2. USE CASE 2 – LEO & GEO OBSERVATIONS

This use case constitutes a very generic usage scenario of SDCs. Thus, its key requirements are reflected in the section on “consolidated key system requirements”.

4.3. USE CASE 3 – LUNAR LANDER

Table 3 Key Requirements for Lunar Lander Use Case 3

#	REQUIREMENT	MINIMUM VALUE
SR-U3-DP-1	Lander (SDC-node) processing capability (float32)	5 TFLOPs
SR-U3-DP-2	Lander (SDC-node) on-board storage	500 GB
SR-U3-LB-1	Single-channel, unidirectional data-bandwidth from SDC-node (lander) to relay-satellite	100Mbps

4.4. CONSOLIDATED KEY SYSTEM REQUIREMENTS (DRAFT)

Final summary will depend on availability of the final system-level-requirements report v2.0

Table 4 Consolidated Key Requirements for all 3 Use Cases

#	REQUIREMENT	MINIMUM VALUE
SR-U1-LB-2	Mothership to ground station data bandwidth (max. 10 min. communication per ground station pass)	500 Mbps
SR-U1-LB-2	Mothership to connected SDC bidirectional data bandwidth (live stream of 5000x5000 x 8 bit x 10 bands every 30 s, plus overhead)	100 Mbps
SR-U2-LB-1	Mothership can concurrently connect to multiple other SDC-nodes	4 other nodes
SR-U1-OA-1	Use and provide standard communication interfaces and protocols	N.A.
SR-U2-OA-2	Dynamically add/remove clients to/from SDC nodes	N.A.
SR-U2-OA-3	Connection transfer from one SDC node to another must not incur any data-loss (i.e., guarantee uninterrupted connectivity)	N.A.

SR-U2-OA-4	SDC node needs to be able to receive/absorb buffered client data in case of link interruption, worth of 10 minutes client acquisition, within a 10-minute connection window. See also SR-U1-LB-2	Unclear
SR-U2-CM-1	Clients may request a high-priority, low-latency data-subchannel for alerting and other low-bandwidth data. These channels will be prioritized.	N.A.

5. SYSTEM ARCHITECTURE (WP4)

Authored by: Jonas Weiss, Patricia Sagmeister, with the help of Ingmar Meijer

5.1. INTRODUCTION & OVERVIEW

According to the technical tender document 7732, this section:

1. Identifies technologies and subsystems required by individual System-Level Requirements
2. Maps system-level requirements to different system-level architectures and their variations.

In the context of this study, we consider the Systems Architecture (SA) mainly as the structure and mode of interaction of subsystems. Specifically, we are interested how data-sources will be connected to data-processing nodes, how the latter are interconnected among others, and how finally the processed data or information makes its way to terrestrial aggregation nodes or to downstream applications and data consumers. After a high-level definition of a "System Architecture", we review similar systems which are already deployed. Then we try to order some thoughts and concepts, also looking for inspiration from terrestrial datacenters to finally derive a dataflow and supporting hardware architecture for a space data center (constellation).

5.2. SYSTEM ARCHITECTURE DEFINITION

System architecture pertains to the strategic design and configuration of hardware and software components, as well as their interconnections and communication protocols, to enable efficient and reliable processing, storage, and transmission of data and information from the source to downstream applications. It involves determining the placement of computational resources, defining the system's logical structure, and establishing the necessary interfaces and protocols to ensure seamless integration and effective collaboration among the interconnected devices and subsystems.

5.3. EXISTING CONNECTED SPACE INFRASTRUCTURE

Space data centres are interlinked, orbiting compute resources which process, store, and relay observation-data from “sensing” client satellites, to either aggregation nodes in space or to the ground, to finally make the insights available to downstream-tasks/users. The architecture of interest is thus composed of the following **system components,** of which we mainly consider 1) and 2), and to some extent 3):

1. The client (satellite), typically with some sort of earth or space observation instrument, which is the data-source.
2. One or several space data centre nodes, mostly on LEO orbits, possibly extended to GEO.
3. A “network” of ground stations. They may be reached through radio frequency (RF) or optical channels. Ground stations may have edge processing capabilities as well.
4. (Public) (cloud) data centres. They may run portions of the algorithms and constitute the interface or access point of the downstream application for the end-user.

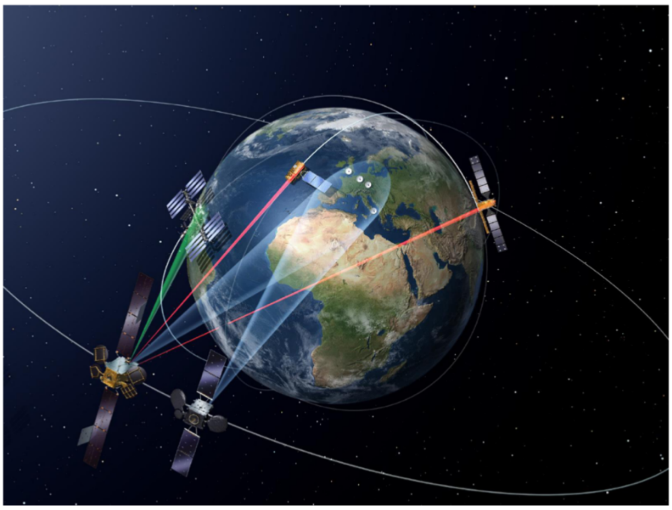


Figure 3 Simplified representation of the concept, using LEO observation and GEO data-relaying orbits (Ref.: GEO segment for Space Data Highway (EDRS)[1])

In the next subsections, we give an overview of existing (large) constellations in orbit, which serve as inspiration for deriving certain aspects of the SDC architecture. An extensive overview of other constellations is provided, e.g., at the link below[2].

[1] https://spaceops.iafastro.directory/a/proceedings/SpaceOps-2021/SpaceOps-2021/1/manuscripts/SpaceOps-2021,1,x1419.pdf

[2] https://www.newspace.im/

5.3.1. EXAMPLE 1: STARLINK CONSTELLATION

Starlink is one of, if not the largest operational satellite constellation. With ~ 4000 operating nodes and a mesh-network architecture, architecture-wise, it resembles a lot our initial vision for a network of Space Data Centers, by already covering the networking aspect. If Starlink and similar constellations will be available to connect to from other spacecrafts, future SDCs may exploit these existing intersatellite network and down-link capabilities. This could help bootstrap SDCs starting with relatively few nodes, focusing on computation, and using existing communication networks and relaxing communication requirements from SDCs.

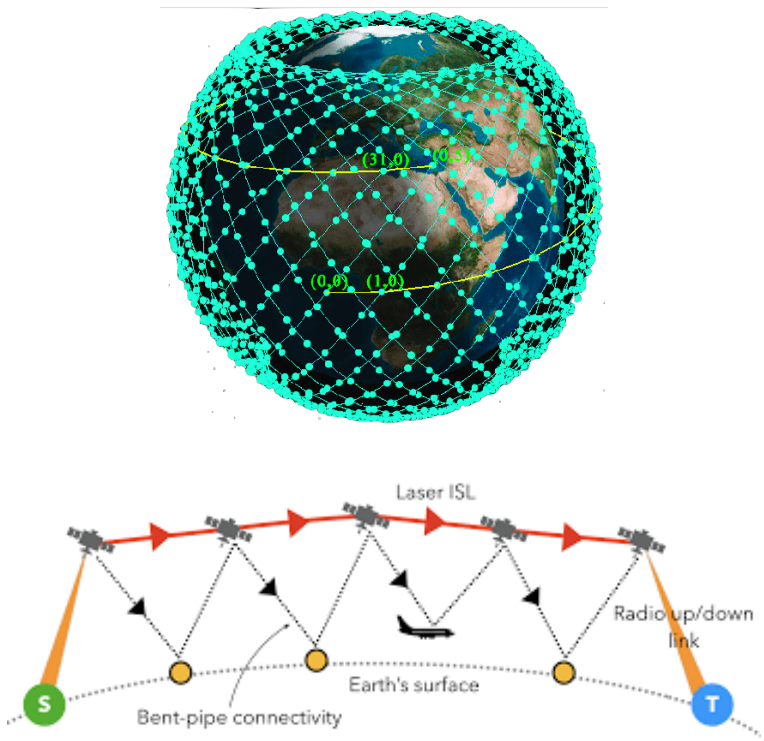


Figure 4 Starlink Constellation "Architecture"

Table 5 Starlink Constellation Summary

# Satellites	4000 (12'000 planned)	Unit
Orbit	550 (340/1110) (3 shells are planned)	km
Size/Weight	227	kg
Power Budget	N.A.	Watt
Product	Internet Connectivity	

5.3.2. EXAMPLE 2: ONEWEB CONSTELLATION

The OneWeb constellation could potentially be exploited similarly as the Starlink one. More research will have to be put into understanding how to best connect and exploit such existing infrastructure, and what constraints will have to be overcome, both, from a legal and economic perspective.

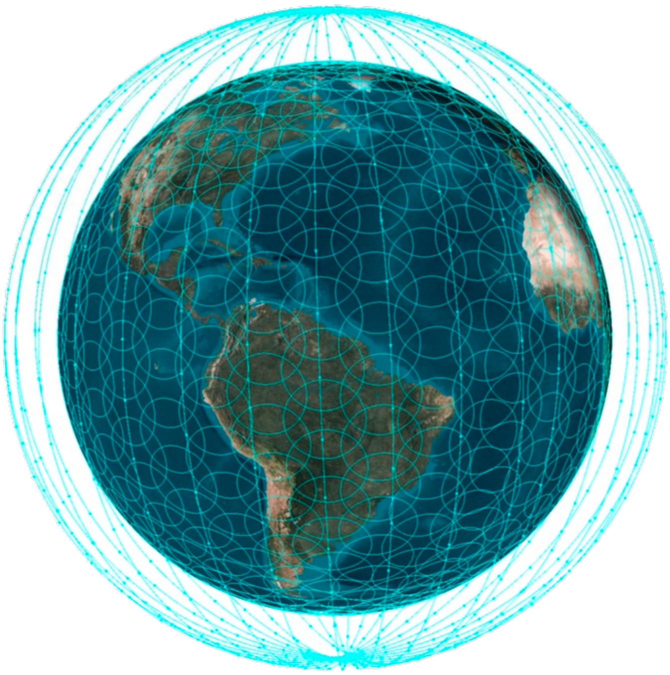


Figure 5 OneWeb Constellation "Architecture"

Table 6 OneWeb Constellation Summary

# Satellites	648/6372 (2000 granted)	Unit
Orbit	800/1200	km
Size/Weight	150	kg
Power Budget	N.A.	Watt
Product	Internet Connectivity	

5.3.3. EXAMPLE 3: GPS CONSTELLATION

The GPS is probably one of the oldest, operational constellations. It is mentioned here more as a reference and not a complement or extension, as its orbit is much higher and high-bandwidth inter-satellite interaction and communication is not a key feature for the service provided by this constellation.

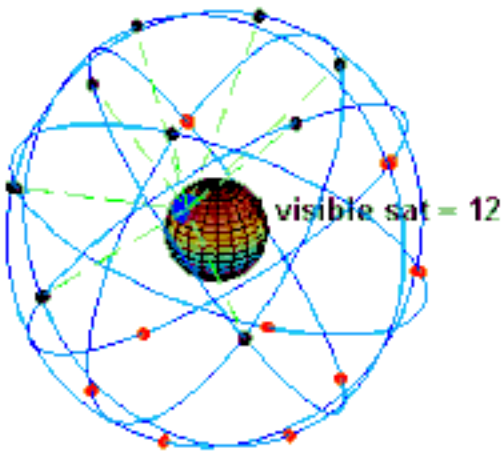


Figure 6 GPS Constellation "Architecture"

Table 7 GPS Constellation Summary

# Satellites	32/38	Unit
Orbit	20 180	km
Size/Weight	1080	kg
Power Budget	N.A.	Watt
Product	Position/Navigation	

5.3.4. EXAMPLE 4: ARTEMIS

The Artemis mission is provided as a reference to an early example for many of the technologies (direct space-to-space links, re-programmability, etc.) that we believe will be key for the proposed SDC concept. Artemis was an ESA telecommunications research satellite that reached GEO 11 years ago. While ownership was transferred to Avanti Communication in 2014, it is still operational and in use.

Artemis demonstrated the first laser data link between satellites in different orbits, established two-way links with a flying aircraft at 50 Mbps, and a UAV flying at above Mach 1. It was the first telecom satellite to be extensively reprogrammed in orbit, provided a data relay for Envisat, and provides links for all of ESA's Automated Transfer Vehicle (ATV) missions to the International Space Station, from launcher separation to docking, deorbiting and, finally, reentry.[3]

Figure 7 ESA ARTEMIS I Mission Satellite

5.3.5. SUMMARY AND GENERAL THOUGHTS

The few largest existing constellations that are in orbit provide global high-speed internet connectivity. To provide uninterrupted global coverage, a relatively dense network of satellites and ground stations is thus required. SDCs too, will require a "relatively" dense mesh, but unlike internet provisioning to the ground, client density will be comparably sparse and the distance between clients and SDC satellites can be up to a few thousand kilometers, thus significantly reducing the required number of SDC nodes in comparison to aforementioned constellations (see, e.g., Eq. 7).

[3] https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Artemis

5.4. CONCEPTS AND IDEAS

As the name Space Data Centers (SDCs) suggests, the core idea of SDCs, and services supplied to clients by SDCs are very similar, if not identical to data centers as we know them from “terrestrial” applications. We can learn from such established installations and only adjust where specific conditions require it. In the following subsections we investigate physical- and dataflow-architectures of terrestrial state-of-the-art. We will mainly focus on the interconnection of SDC nodes, as the node-design itself is discussed in Section 9 of this document. Subsequently we will discuss how existing schemes could potentially be mapped to datacenters in orbit.

5.4.1. EXISTING DATA CENTER ARCHITECTURES

Before looking into architecture details, we shall create a common terminology reference and define the use a few key terms in the context of SDCs and data centers (DCs), deriving them from terrestrial DCs[4].

Terrestrial Term	**SDC Equivalent**
Data Center	SDC Constellation
DC Rooms	SDC Ring (SDCs on same orbit)
DC Pods (smallest self-contained DC unit)	SDC Node (Single satellite)
DC Rack (of servers, switches, etc.)	SDC Pod (one of up to several on-board “racks”)
DC Server	SDC Server

Looking up architectures of existing data centers yields figures such as Figure 8, Figure 9 or Figure 10. Besides the actual compute-, memory- and storage hardware, the key-elements and performance differentiators of many data centers are their networking components, switches, routers, and interconnection architectures such as depicted in Figure 10.

Figure 8 depicts a data center pod from a *softlayer* data center. Towards the network attached Pods B & C, e.g., Pod A is visible through servers; Dedicated (hardware/bare metal) and **virtual servers**. Otherwise, routers and switches dominate the figure, highlighting the relevance of pod-to-pod and “pod-to-the-world” interconnection.

[4] Specification of Modular Data Center Architecture, Neil Rasmussen, Schneider Electric (https://www.se.com/ww/en/download/document/SPD_NRAN-7ZE9R3_EN/)

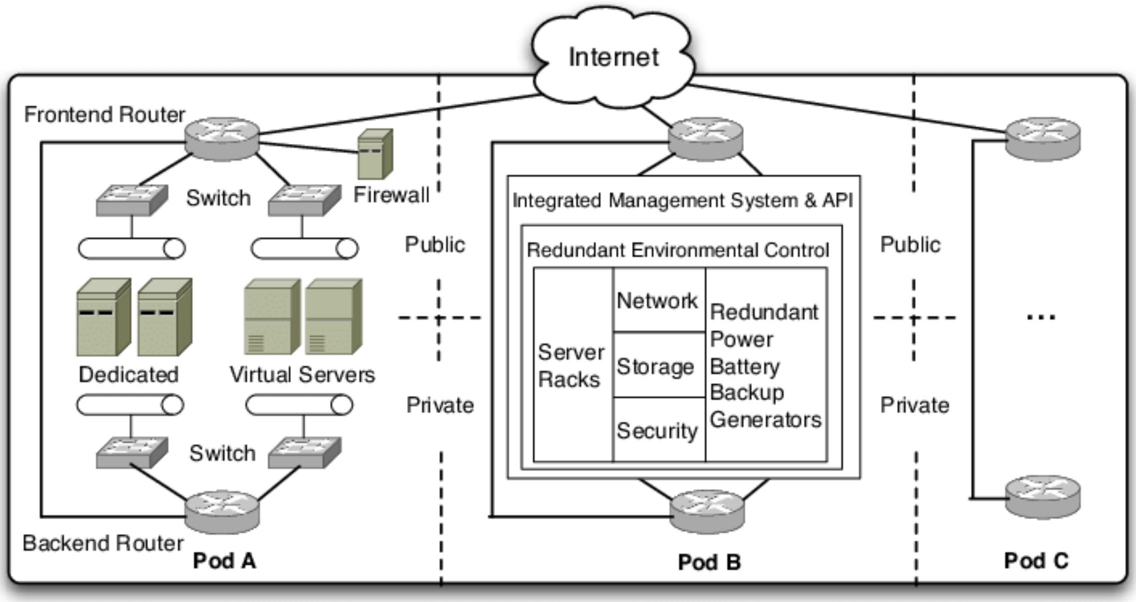


Figure 8 Softlayer Data Center Pod Design

An extended view of Figure 8 is shown in Figure 9, where additional infrastructure and power challenges are included. As discussed earlier, several racks can be grouped into a pod. Also here, the **virtual machines** are shown as **the application facing lowest level services** provided by the DC.

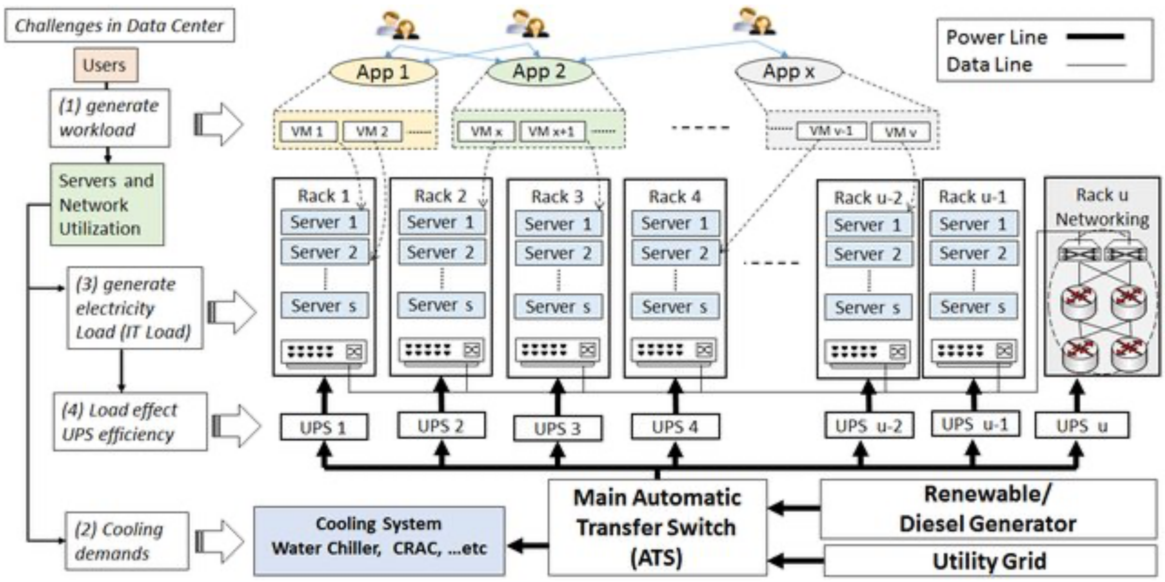


Figure 9 Data center infrastructure design and power delivery paths

5.4.2. EXISTING DATA CENTER NETWORK ARCHITECTURES

An important differentiation also needs to be made on whether the data center will run many small, independent workloads (webservers, online-stores etc.), e.g., through virtualization techniques like VMs (virtual machines), docker containers or both, or if very large workloads need to be partitioned and distributed over many nodes, thus high-performance computing (HPC) performance will be key. In the first case, relatively relaxed requirements on inter-core bandwidth are possible, as many workloads will effectively share the same physical compute core and through time-multiplexing have their VMs perform their tasks. In the latter case, where single workloads occupy many physical compute cores and reside in physically separated memory, inter-core and inter-node/pod communication become critical for efficient workload execution. Consequently, also DC networks must account for such different requirements.

For SDC **workloads** at an early and intermediate time-horizon, it’s probably safe to assume that they **will always fit into a single SDC node** (satellite). For future applications that want to exploit distributed data, either for model inference or even for learning, schemes such as “federated learning” are currently explored that are specifically tailored to provide deep insights without having to tightly interconnect the participating compute/data nodes. In consequence, even though there is a trend towards hyperscale DCs on the planet, which may come close to HPC capabilities, **for SDCs NO HPC workloads are currently foreseen**, which will relax network requirements to, what we believe, quite practical levels.

Comparing existing DC network architectures such as depicted in Figure 10, unveils the hierarchical nature of all of them.

Mesh networks are generally considered **higher performance** with **excellent fault tolerance**, but more **complex to manage** and configure, specifically when new nodes need to be added.

Multi-Tier Models are very **scalable** and **easier to manage**, but **less reliable,** e.g., within layer dependencies in the case of single failures.

Mesh Point of Delivery (Mesh PoD) is a highly interconnected architecture that provides good redundancy and can ideally exploit data locality within sub-network. It is more often implemented in smaller DCs.

Super Spine Mesh Network combine the connectivity and redundancy advantages of mesh networks on top of a leaf-spine architecture. They seem to be not very common and are shown **for reference only.**

If we jump forward to Section 5.4.5 or Section 5.4.6 the concept of pseudo-static and dynamic connections between SDCs is introduced in detail, describing how SDCs on the same orbit are connected and how different SDC orbits are dynamically interconnected. Accounting for the "ring-like" structure of SDCs on the same orbit, the **"Mesh Point of Delivery" network** as depicted in Figure 10 on the bottom left **seems intuitively a good fit** to map the intrinsic SDC architecture to existing network architectures. While for SDCs a full mesh as depicted in Figure 10 will not be possible, it would still allow to exploit an existing architecture with minimal changes and additional constraints, thus profit from deployment schemes and experience from the latter.

Whether it is an optimal choice will be discussed later, also as a function of how the physical boundaries of the SDC (constellation) are put, with respect to the mapping to the network architecture.

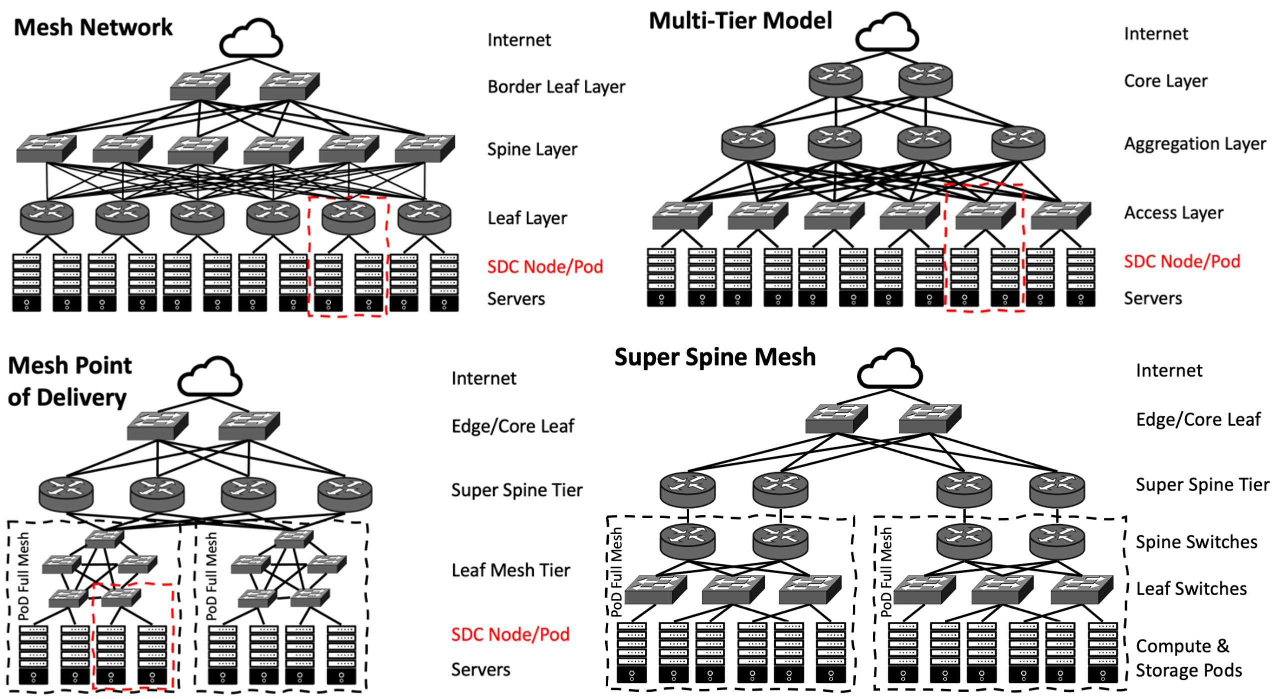


Figure 10 Data center network architectures
*(**Mesh Network** for high connectivity, performance and redundancy, **Multi-Tier Model** for reasonable redundancy and excellent scaling and **Mesh Point of Delivery** for a combination of the latter two, mostly for smaller installations and for reference only: Super Spine Mesh Architecture)*

5.4.3. DATA(FLOW) ARCHITECTURES

The previous sections were mainly referring to hardware related/constraint architectural considerations. The Data Flow Architecture, while of course being affected by the hardware layout, is focusing on software related data-aspects. Generally, it describes a software system as a series of data-independent transformations on consecutive pieces of data that are entering the system. After the data has flown through the modules of the system, the transformed data either leaves the system or is being stored. Between the software building blocks, I/O streams, buffers, pipes or other connections are used. Dataflows can easily be visualized using a graph representation. Their main objective is to foster system planning and understanding towards reusable components/modules.

In Data Flow Architectures it is often differentiated between module execution sequences: *Batch Sequential*, *Pipe and Filter* and *Process Control* data flow architectures. The purpose of the SDC is to provide a flexible data ingestion, storage, and processing infrastructure, to which clients can upload their own workloads. While this would call for a "Dataflow Agnostic" architecture, it is likely that most relevant **workloads on SDCs will either be of *Batch Sequential* or *Pipe and Filter*** type.

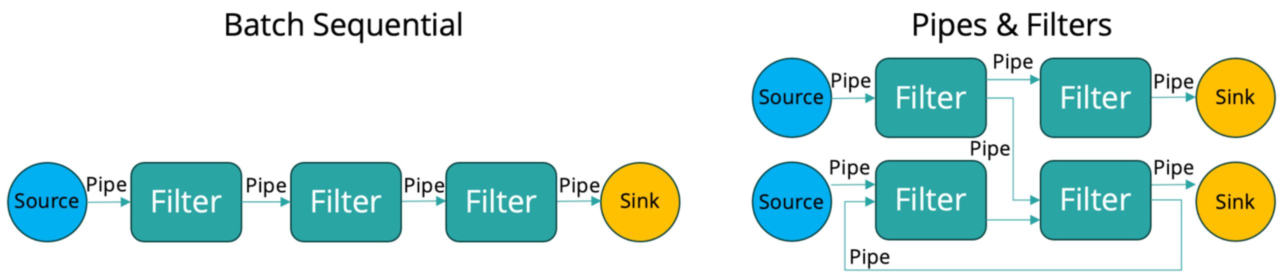


Figure 11 Data Flow Architectures (Batch Sequential & Pipes & Filters)

Batch Sequential describes a highly expected user scenario, where data is received from a client satellite, is processed chunk wise, e.g., for tasks like wildfire- or flood- detection, and the results, e.g., fire/flood coordinates or outlines are sent to the ground station for further distribution.

Pipes and Filters depicts a more sophisticated processing scenario, either on a single SDC node or distributed over several, where specifically data from different sources are co-processed. The sink can either be a consumer on the ground, or buffering storage for further processing on SDCs.

A possible overall SDC Data Flow Architecture, additionally accounting for an SDC Gateway and an ingestion stage is depicted in Figure 12.

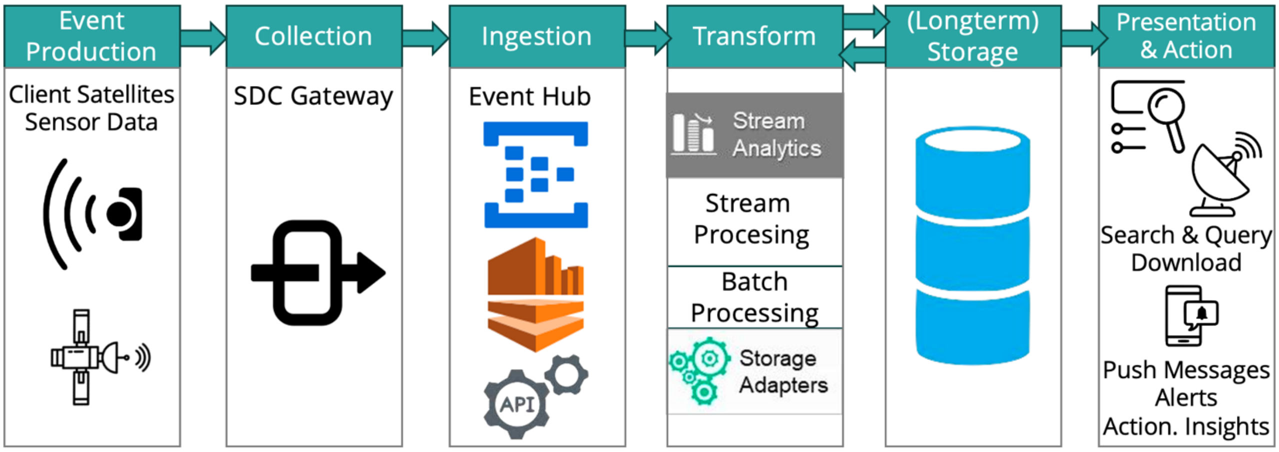


Figure 12 SDC Data Flow Architecture

This pipeline will help derive requirements and implementation of the software and operating system stack in subsequent chapters.

5.4.4. DATA SECURITY

Securing Data in cloud environments and datacenters is one of the most critical tasks of Cloud Service Providers. The NIST Cybersecurity Framework 2.0 [6] states on page 37: ”Data Security (PR.DS): Data is managed consistent with the organization’s risk strategy to protect the confidentiality, integrity, and availability of information”. While confidentiality addresses the secrecy of any data during its whole lifetime, integrity ensures the authenticity of data. This includes the protection of

- data at rest (PR.DS-01 [6]),
- data in transit (PR.DS-02 [6]), as well as
- data in use (PR.DS-10 [6]).

Data at rest can currently be protected through various standardized encryption mechanism, such as symmetric or asymmetric encryption. Data in flight in state-of-the-art data centers and cloud environments is protected through some sort of TLS or mTLS (mutual TLS) including VPN (Virtual

Private Networks) or Ipsec mechanisms. Fully homomorphic encryption (FHM) seems to be a promising technology to allow computations on fully encrypted data in use, where the trust boundary excludes memory and even the CPU/processor cores. As FHM is still in its infancy, it is not yet used in commercial data centers or cloud environments.

Instead, confidential computing is offered via TEEs (Trusted Execution Environments), such as Secure Virtual Machines or enclaves. All TEE technologies rely on hardware / platform support and restrict the trust boundary to the CPU. They ensure the isolation of applications during runtime as well as the confidentiality of their respective used data and meta-data, located in memory. However, TEEs are not standardized, and every platform manufacturer offers its own TEE technology with its proprietary integrity and checking and attestation mechanisms. Examples are Intel DCAP / Intel TDX, Arm TrustZone, AMD SEV, IBM Secure Execution, just to name a few. Consequently, TEE technologies used in commercial data centers and cloud environments do not provide any inter-platform compatibility for data in use protection, nor integrity checking.

In September 2022 the National Security Agency announced in its cyber security advisory the Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) [8] . This document defines the transition timeline to the new quantum safe public-key and symmetric-key algorithms. According to this timeline, which is depicted in Figure 13, software and firmware signing must use CNSA 2.0 algorithms exclusively by 2030, cloud services by 2033, and equipment providers by 2033. This timeline also affects data security for SDCs. First, the realization of an SDC is projected around 2030 where all equipment providers must be compliant to CNSA 2.0. Second the shelf life of the generated data and insights on board of an SDC is way beyond 2033.

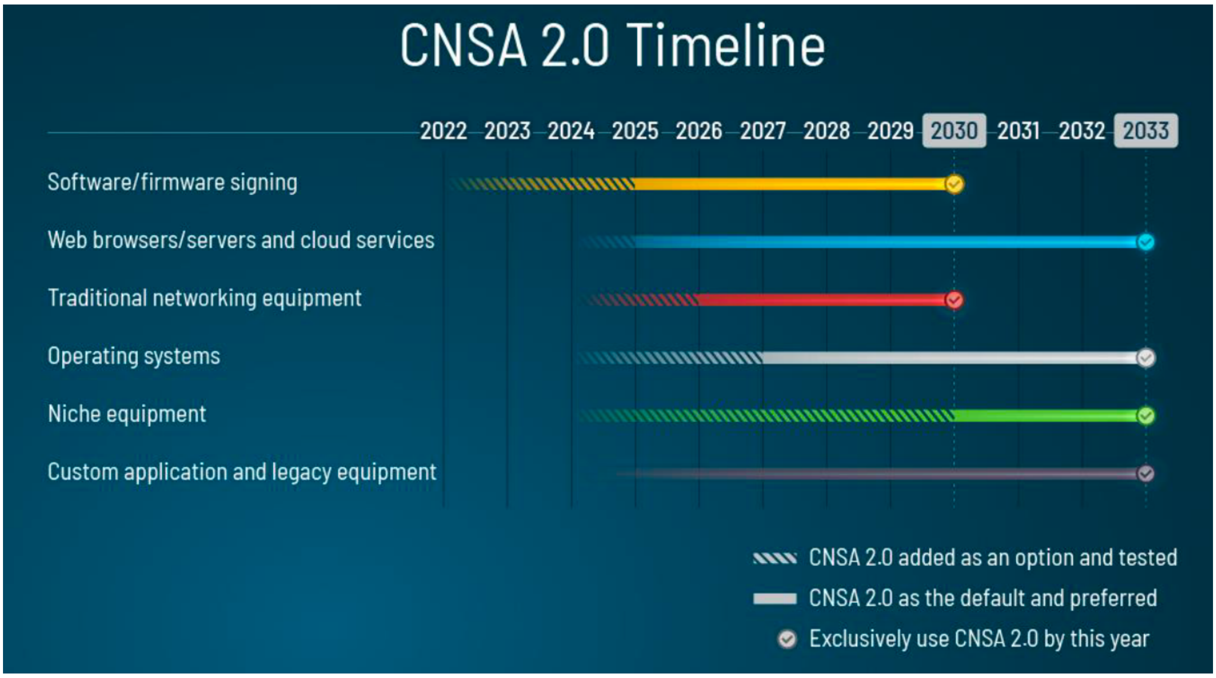


Figure 13 CNSA 2.0 Transition Timeline to Quantum-Safe Algorithms

Third, harvesting of data now is an existing threat for SDC generated data and insights and has a significant impact on the security level of data, as shown in Figure 14. Most of the symmetric cryptographic algorithms and all asymmetric encryption algorithms will be broken, allowing the decryption of harvested data that is not QSC-hardened (Quantum-Safe Cryptography) now. So, data that is protected today is already vulnerable.

	Name	Function	Pre Quantum Security Level	Post Quantum Security Level
SYMMETRIC	AES-128	Block cipher	128	64 (Grover)
	AES-256	Block cipher	256	128 (Grover)
	Salsa20	Stream cipher	256	128 (Grover)
	GMAC	MAC	128	No impact
	Poly1305	MAC	128	No impact
	SHA-256	Hash function	256	128 (Grover)
	SHA-3	Hash function	256	128 (Grover)
ASYMMETRIC	RSA-3072	Encryption	128	Broken (Shor)
	RSA-3072	Signature	128	Broken (Shor)
	DH-3072	Key exchange	128	Broken (Shor)
	DSA-3072	Signature	128	Broken (Shor)
	256 ECDH	Key exchange	128	Broken (Shor)
	256 ECDSA	Signature	128	Broken (Shor)

Rule of Thumb:

Security Halved

Quadratic improvement in brute-force attacks on symmetric encryption like AES

Security Broken

Exponential improvement in brute-force attacks on asymmetric encryption schemes like RSA

Figure 14 Impact on security level Pre versus Post Quantum

The NIST Interagency Report 8270 [7] maps the NIST Cybersecurity Framework 2.0 [6] to commercial satellite operations and establishes with this a dedicated cybersecurity profile. While protecting data at rest (PR.DS-1 [7]) and data in transit (PR.DS-2 [7]) are included as subcategories, ensuring the secrecy of data in use is missing as only current satellite operations are targeted. However, when looking at SDCs, the protection of all data in rest, in transit, and in use has to be ensured.

Additionally, the profile does not include the transition to quantum-safe cryptographic algorithms as answer to the "Harvest now – Decrypt later" threat. Therefore, the following QSC-transition, depicted in Figure 14, is suggested for SDCs to ensure the protection of today's generated data also after Q-Day.

		Time Domain	
		Before Q-Day	**After Q-Day**
	Existing Threat	Harvest data now	Decrypt harvested data
Usage Domain	Data at rest	• Ensure to use AES with 256 bits • Do not wrap AES keys in legacy asymmetric public keys	• Add QSC or hybrid certificates for authentication • Use a QSC PKI
	Data in transit	• Ensure that key exchange is using QSC or hybrid algorithms • Use a symmetric AEAD cipher with at least 256 bits: AES-256-GCM, ChaCha20-Poly1305 • Applies to ALL communication	• Add QSC or hybrid certificates for authentication • Use a QSC PKI
	Data in use	• Start QSC-hardening of infrastructure as soon as technology becomes available	• QSC-harden your infrastructure up to the platform level FW, boot process, hypervisor, OS & cluster

Figure 15 QSC Transition Recommendation

5.4.5. ORBITS AND INTER-ORBIT COMMUNICATION

Clearly, the physical architecture of a data center is tightly interwound with the network architecture, thus some overlap and dependency of this section with Section 6 is unavoidable and even necessary.

If we consider a single SDC node as the equivalent to a data center Pod, the physical architecture will strongly be governed by orbital physics and the possible network topologies given by orbital parameters of the SDC nodes.

Theoretically, every SDC node could be on a completely independent orbit and clients could randomly connect and drop connections with them and within the network of SDCs. The SDC nodes would be “pseudo randomly” connected to each other, which for certain constellations may even result in “beat frequency like” high-density to low-density satellite coverage for specific areas, which we consider suboptimal.

We therefore assume that a purpose built **SDC constellation will consist of different groups of SDCs on the same orbit,** with multiple such **SDC rings interconnected** at least near their orbital crossing points. This allows to better predict network and service availability and mostly reduces the complexity of dynamically interconnecting the SDCs on the different orbits with each other.

With this assumption and from Figure 19 e.g., it becomes clear, that there will be two different communication hierarchies of SDC nodes. In the context of Section 6 (Network Architecture & Management), we refer to them as pseudo static and dynamic. All SDC nodes on one specific orbit form a **pseudo-static** cluster or rather a **ring of pods** with all other SDCs **on the same orbit**. All these rings then have realistically 2-4 dynamic connections with SDCs on other rings (see Figure 16).

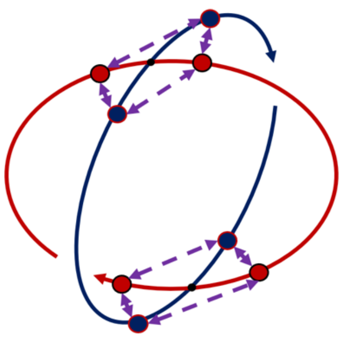


Figure 16 Dynamic connections between two rings of pods (SDC nodes) at their crossing points

We can compare this setup with the constellations of Starlink and OneWeb (Sections 5.3.1 & 5.3.2). While the Starlink constellation orbital crossing "evenly" distributed across the globe, OneWeb opts to have all crossings between the orbits in the polar region, creating a very high coverage within these regions. The exact motivation for this is not known, it maybe that there, since little coverage on the ground is needed, capacity of the satellites can be drawn from ground-coverage to inter-orbit workloads, or that almost parallel orbiting satellites from north-to-south can continuously communicate east-west and vice versa.

From a service perspective to client satellites, the 2 constellation arrangements may not be significantly different. Intuition however tells that likely the Starlink constellation may be better suited to "load balance" communication infrastructure on the SDC nodes. Thus, if required in the following, **we assume rather a Starlink like constellation** than a OneWeb type and for **inter-orbit communication**, we suggest using the emerging high-bandwidth and highly dynamic orbital terminals, instead of RF links (see also Figure 19)

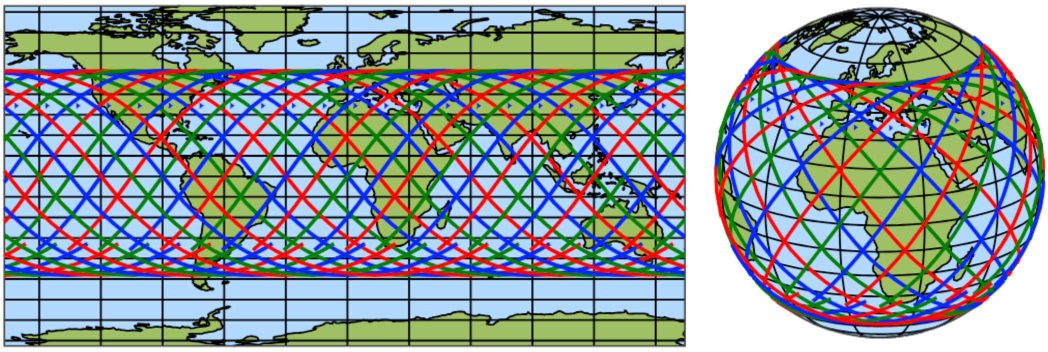


Figure 17 Potential constellation with 24 orbits
(Classical orbital elements: a= 7000 km, ecc = 0, inc. = 53 deg., raan = 0 deg., argp = 0 deg. and nu = 0 deg.)

5.4.6. INTRA-ORBIT COMMUNICATION

To achieve long distance links (few thousand kilometers), account for data-privacy (i.e. links between satellites must be considered as "private" as the links between racks in a data center and to not congest the RF spectrum unnecessarily, **directed, free space optical (laser) links between neighboring satellites** are foreseen (recommended). As shown in Figure 18, it is not compulsory to have equidistant satellites on the same orbit, but for practical reasons would be desirable. To estimate benchmark figures/requirements, also in the simulation tool from Section 10, **we do assume equidistant distribution of satellites on a single orbit** though.

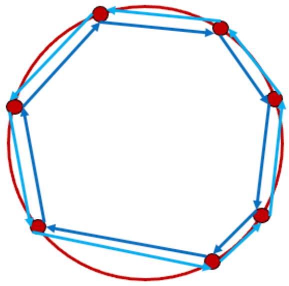


Figure 18 Intra-Orbit Communication will be bidirectional, satellite spacing does not have to be equidistant (fully populated), but to "close" the communication ring, all satellites need to be within the range of the laser terminals.

While using free-space optical links provides the means for very long range, links virtually not interfering with each other at very high bandwidth, they may be constraint by solar straylight. See Section **Error! Reference source not found.** for more details.

5.4.7. SUMMARY OF ASSUMPTIONS

1. Purpose built SDC constellation will consist of different groups of SDCs on the same orbit.
2. Multiple such SDC rings are interconnected exclusively near their orbital crossing points.
3. All SDC nodes on one specific orbit form a pseudo-static cluster or rather a ring of pods with all other SDCs on the same orbit.
4. The SDC constellation will assume rather a Starlink like configuration than a OneWeb.
5. For inter-orbit communication, we suggest using optical laser terminals/links.
6. For intra-orbit communication, we suggest using optical laser terminals/links.
7. Equidistant distribution of satellites on a single orbit for simplifying initial calculations.

5.5. GENERIC SDC ARCHITECTURE ASSUMPTIONS

To create a scenario which is easy to calculate and intuitive to understand, we make the following assumptions, with illustrations in Figure 19 :

i. SDC will be mostly market driven, providing services for commercial application and only secondary to purely scientific missions.

ii. Because of (i) we suggest orbits that mostly cover inhabited or resource wise commercially relevant areas, e.g., not fully covering polar areas.

iii. We assume trailing constellations of SDCs on the same orbit, with a time-lag corresponding to less than or equal the maximum possible inter-orbit communication distance.

iv. SDCs on the same orbit will always maintain a connection with their immediate neighbors.

v. Because of iii) & iv), any crossing orbits will always be (network) connected at their two crossing points, irrespective of the number of orbits.

vi. SDCs make the most sense when continuous availability to clients can be provided. For this, a “virtual” grid of SDCs is proposed (see Figure 19).

vii. From vi) SDCs will/may be tightly interconnected in a mesh-architecture.

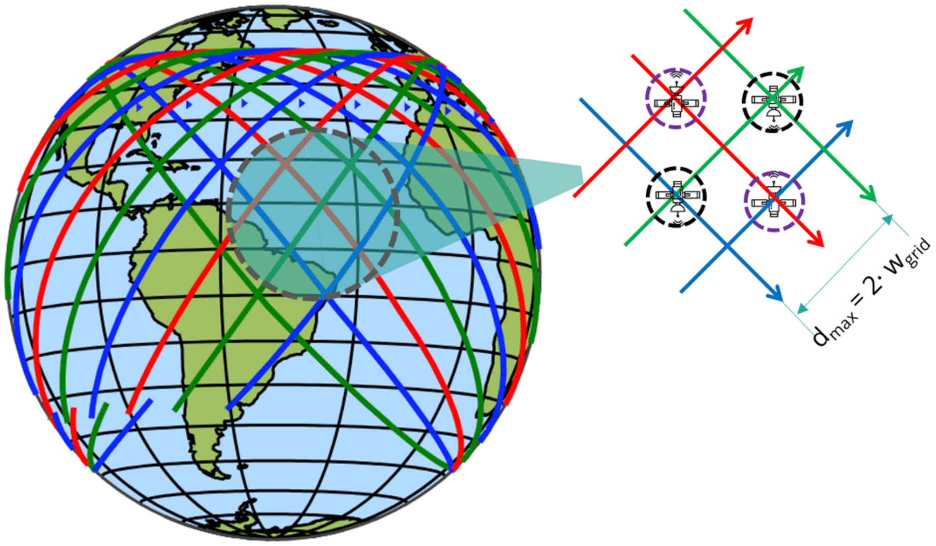


Figure 19 Proposed SDC orbits for best serving clients over mostly inhabited area (leaving out polar regions – 53 deg. inclination -> Starlink) Maximum SDC=SDC (communication) distance is 2x maximum distance "between" orbits around the equator.

Given the maximum communication distance d_{max} between two SDCs (Figure 19) the maximum "equatorial orbit-distance" w_{grid}, can be derived with Eq. 3.

The number of satellites per orbit n_s and minimum number of orbits n_o can be approximated with subsequent equations based on the available maximum length of the communication links d_{max}:

$$w_{grid} = d_{max}/2 = 2000\ km \qquad \text{Eq. 3}$$

$$n_o = 40'000km\ /\ w_{grid} = 20 \qquad \text{Eq. 4}$$

$$n_{s_{perOrbit}} = 40'000km\ /\ d_{max} = 10 \qquad \text{Eq. 5}$$

$$n_s = n_o \cdot n_{s_{perOrbit}} = (40'000km\ /\ d_{max})^2 \cdot 2 = n_o{}^2 \cdot 2 = 200 \qquad \text{Eq. 6}$$

As an example, assuming a communication distance between SDC nodes d_{max} = 4000 km, 200 satellites will be required for the constellation to always be "tightly" connected.

$$n_s = (40'000km\ /\ 4000km)^2 \cdot 2 = 200 \qquad \text{Eq. 7}$$

Countless equation such as above, are implemented either within the main visual basic code (VBA) in the spreadsheet macros, or where applicable also directly in respective spreadsheet cells.

5.6. ARCHITECTURE SUMMARY AND OUTLOOK

We aim at drafting a “general-purpose” system in this work. Thus, flexibility to adapt to, and to fulfill different requirements is key. In this regard, a simple and deterministic structure that allows to connect to and interact with SDCs by means of predictable and repetitive patterns, will make it easy for early and later clients to plan for and consume services of SDCs. While there are different viable options and variations to such an SDC constellation architecture, we believe that a close to even distribution of nodes, similarly to StarLink and OneWeb, will serve the purpose of SDCs and their customers best.

In an early deployment phase, the number of SDCs should match the maximum available long-range (optical) communication distance available, while for follow-up phases, where economy of scale may be exploited, smaller, cheaper and lighter communication terminals maybe installed on additional SDC satellites, which either fill gaps on existing orbits or complement the SDC “grid” towards a finer mesh by occupying additional orbits.

6. NETWORK ARCHITECTURE & MANAGEMENT (WP6)

Authored by: Dinesh Verma, Sagar Tayal, Jonas Weiss

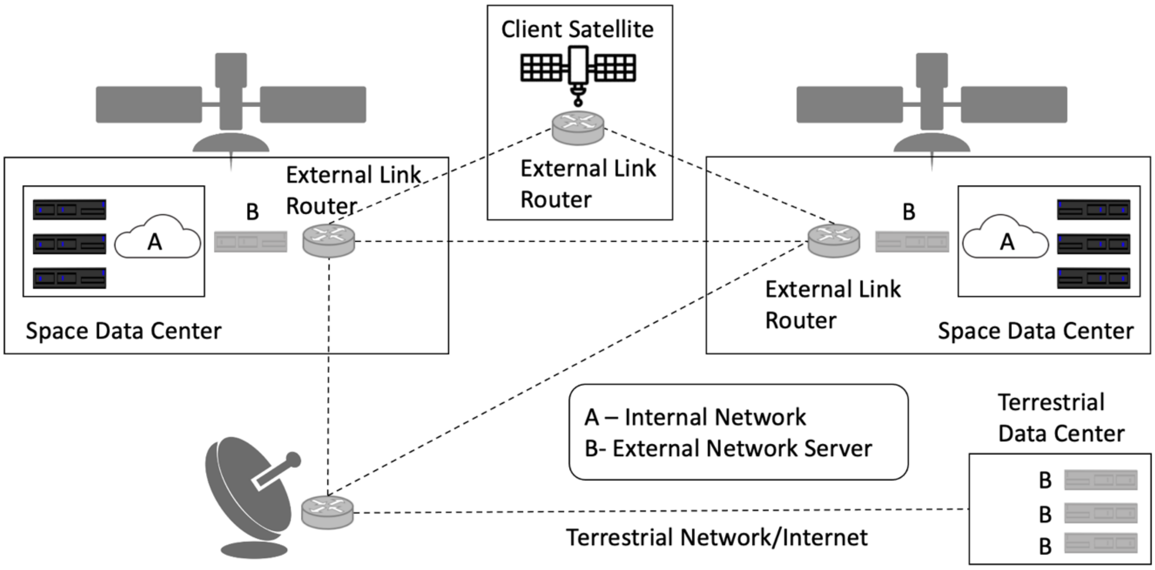


Figure 20 Space Data Center Network Architecture Overview, with 2 SDC nodes and 1 client exemplarily

6.1. GENERAL CONSIDERATIONS & ARCHITECTURE

To present the network architecture for the space data center, we want to initially present an overall, higher-level architecture from the network-centric perspective of the space data center. In Figure 20, two satellites with their space data center payload and connectivity to the ground are shown. There are two types of networks, and their architecture needs to be considered for the space data center. The first one is an internal network which interconnects different elements of the space data center (node), and the second is an external network which interconnects different satellites and terrestrial network together. The network architecture must consider the capabilities of both the internal and the external network for the operation of the overall system.

To efficiently exchange data and orchestrate/distribute heavy workloads between SDC nodes, e.g., through means of free-space optical links, we expect involved routers to be more capable than the ones that communicate with ground-stations or acquire data from client satellites. We thus differentiate between two external network routers: **Dynamic routers**, between ground station or client satellite and SDC nodes, and **quasi static routers**, between SDC nodes (see Figure 25 and

Figure 26). Dynamic and static routers provide the same router functionality but are optimize for different data volumes and static or changing connectivity requirements.

Network A (Internal Network): Interconnects elements of the space data center pod together and is referred to as the internal network.

Network B (External Network): Interfaces with the external network and may provide similar communication capabilities such as provided by Starlink, OneWeb, etc., and may even connect to the latter networks in the future. The external network server is used as the access device for the external network within the SDC satellites and consists of network optimization software as described later. The network optimization software would also consist of a counterpart on the terrestrial data center which contains the same optimization software as well.

From the system design perspective, the internal data center network within the satellite should be modeled in a manner identical to that of the traditional data center networks (see Figure 21 top left part). For the internal operation of the machines within the space data center, the design of a data center network can be replicated within the satellite subject to the constraints of weight and size. In this architecture section, **we primarily focus** on the architecture of the network optimization software **on the external network server**.

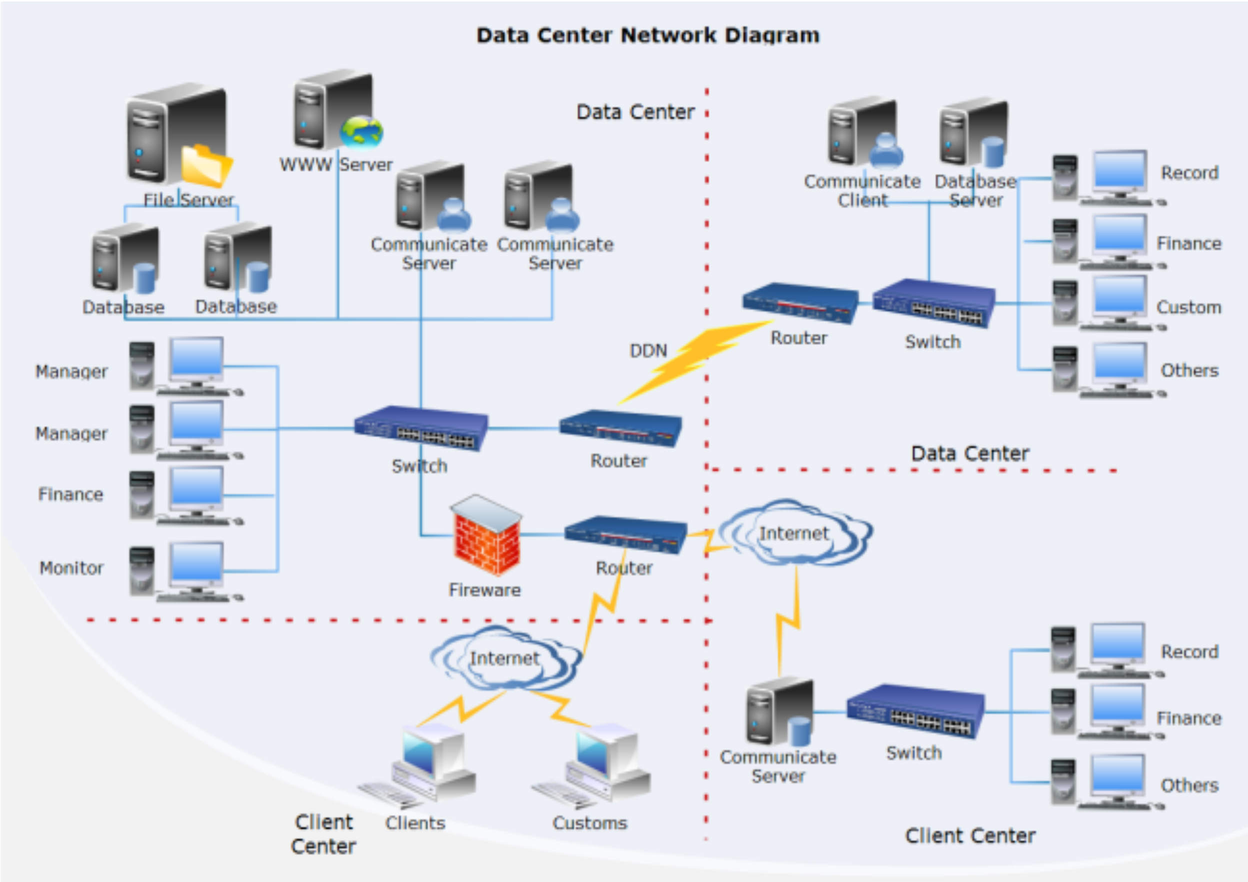


Figure 21 "Standard" Terrestrial Data Center Network Architecture Overview

With network connectivity providers such as Starlink and OneWeb that provide Internet connectivity, the external network software could either connect to their router using the Internet Protocol (IP) at the network level or connect to ground stations in a similar fashion.

Specifically, we recommend the following components for the external network server:

1) **Network state monitor** monitors the state of connectivity with the ground station and other satellites and makes it available to other components.

2) **Disruption tolerant networking** provides for a way to hold IP packets targeted for destinations during the periods the satellite may not have connectivity to other satellites, which are released when connectivity is re-established.

3) **Ad-Hoc Mesh Routing** provides for routing of information from the satellite towards the ground station or in the reverse direction. Various protocols for ad-hoc mesh routing have been proposed in the literature, but given the constrained topology of the satellite infrastructure, a directed tree-based approach can be used effectively and is recommended.

4) **Network Data Compression** examines the content that is being transferred and optimizes it to minimize the amount of data that needs to be transferred. Techniques for network data compression include approaches such as byte-caching and coreset enablement.

5) **Data Caching Proxies** proactively retrieve anticipated data from the other parties when connectivity is maintained and can serve external data to local clients in the space data center.

6) **Customized network Services** support services such as Domain Name Server and file transfer services need to be reimplemented on the external network server so that they can retrieve data needed by them when external connectivity is available and work off the cached content when connectivity is not available.

The combined effect of these software services is the creation of an external Internet connectivity which looks and feels like the traditional network connectivity to applications running on the space data center. From the perspective of those applications, there need not be any awareness of the fact that they are in a specialized space network from a connectivity perspective.

6.2. NETWORK DESIGN CONSIDERATIONS

As for all system components, the discussion of the network architecture shall help assess **module cost, module weight and module energy consumption** as a function of architectural decisions and performance requirements, i.e., input parameters, with the following considerations:

- We **only consider Network B** (see Figure 20). Network A will be covered in Chapter 9.
- We pursue a hierarchical approach in which the network model consists of different links, which are based on the same generic link-model but with different parameters.
-

While RF links could also be used for Network B, it is expected that for SDC applications mostly high-bandwidth optical links will be exploited, overcoming many of RF-links limitations.

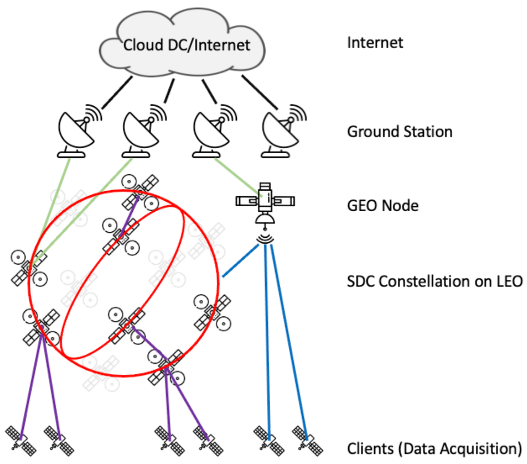


Figure 22 Any point-to-point link between satellites or between satellite and ground station can be modeled as derived below. Only terrestrial links are NOT considered here, as they will likely be fiber/cable based.

While cosmic radiation or solar flares (generally space weather) may impact link quality und thus available bandwidth, such events are too sporadic to be modeled in this context. However, straylight from the sun is also known to affect link performance. Figure 23 shows how with a given receiver angular sensitivity to solar radiation, a specific sector of the orbits will be affected. All “full” orbits will be affected all the time, as is depicted in Figure 23.

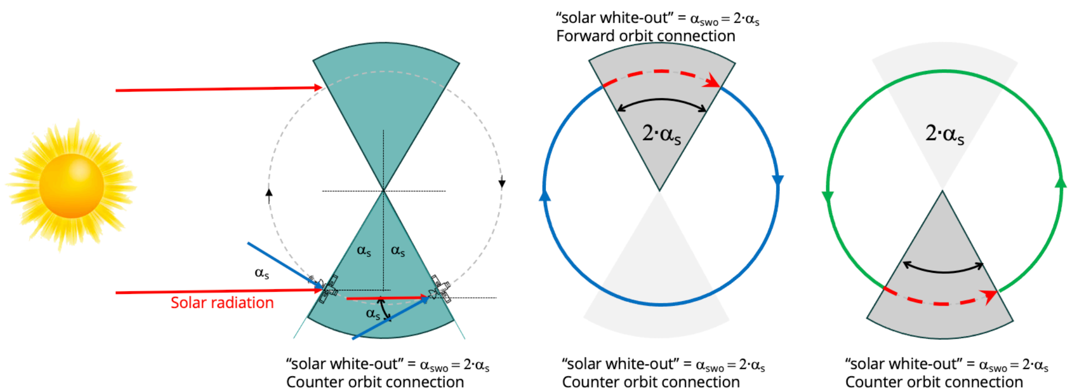


Figure 23 Stray light sensitivity of the optical receiver affects link performance. If the receiver faces the sun with an incident angle below the “sun angle”, bandwidth will start deteriorating. For any realistic scenario, this will only happen in one direction of the link … either counter or forward orbit wise, but of course affect link bidirectionality.

Part of the network optimization will thus include the routing direction within an orbit to avoid the solar straylight affected bandwidth limited section, at least for high-bandwidth payloads. Direct links from client satellites to SDCs may equally be affected, depending on their position. If buffering/caching is not acceptable, rerouting to alternative SDC nodes is an option (see also Figure 19). If not, these circumstances will help define the minimum network caching memory-depth. It has been shown that sun angles as small as 2° are possible for deep space missions [3] , however, the SDA OCT Standard T0/T1 specifically excludes any boresight angle (receiver and transmitter side) that is smaller than 30° with respect to the sun. Thus, outside these 30°, no performance impairment can be expected. Within the 30° however, terminal behavior is normally not defined.

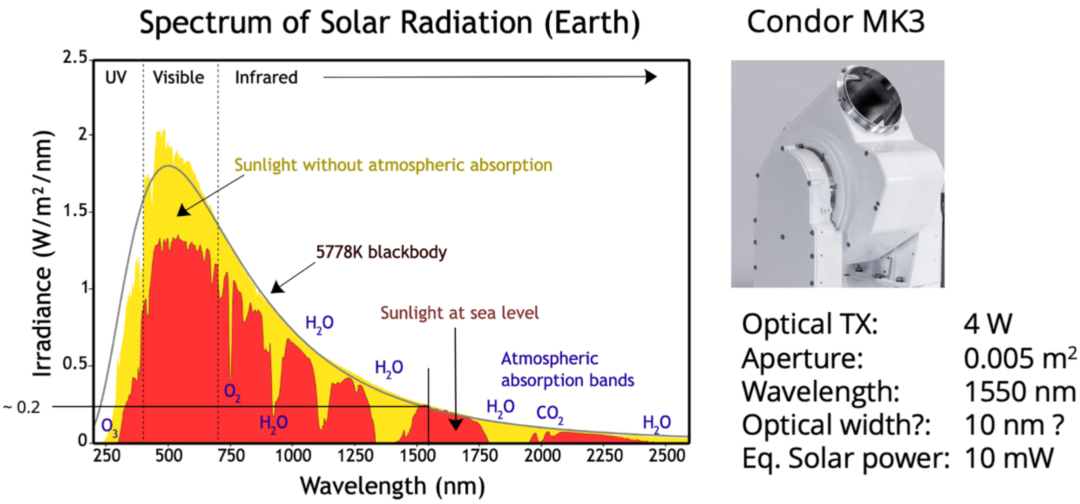


Figure 24 Solar radiation power density. Depending on communication distance and thus how much incident light from the transmitter signal is expected at the receiver, the signal to (solar) noise can be computed. For the Condor MK3 terminal e.g. a laser-line filter with 10 nm spectral width would still let pass 10 mW solar radiation, which for long-range applications will likely be challenging.

We thus assume that the terminal is completely shut down for protection if operated within the 30°. A gradual link degradation, e.g., following a gaussian distribution would also be a valid assumption, but which for reasons of keeping it simple, we will not pursue here. We can thus compute worst case delays if data will have to be propagated into the opposite direction, or caching memory requirements, if communication will be suspended while passing through the 30° solar sector of the orbit.

Table 8 lists the communication link input parameter, which will have to be provided either by a preceding model or by user input/selection. Please see also Section 10 for implementation details (based on limited published information the cost-per-bandwidth-per-distance and power-per-bandwidth-per-distance key performance metrics were derived/approximated to be used in the model).

Table 8 Link Model Input Parameter (i.e., driving values) and Defaults (See footnotes for superscripts)

Symbol	Description	Unit	Default Value
tp_{link}	Bandwidth required, e.g., by workload	Mbps	800[1]
bw_{term}	Bandwidth available from 1 terminal	Gbps	2.5[2]
l_{link}	Maximum distance of a link to still work properly.	km	6500[2]
α_s	Boresight angle towards the sun at which the optical terminal "stops" working (within specs).	degree	30[3]
n_{stream}	How many input data streams will be aggregated to the output data stream		4[4]
p_{router_xx}	Router power per bandwidth (or fixed)	W/Gbps	5[5]
c_{router_xx}	Cost per router	Euro	1'000[6]
v_{orbit}	Speed of a satellite on the orbit	km/s	7.8
S_{orbit}	Length of orbit	km	40'000
d_{max}	Spacing of satellites on the same orbit	km	4000

1) Required by use-case. E.g. 10 Hz 10 MB images 2 8 bit = 800 Mbps
2) Derived from system architecture. Feasible values e.g. by Mynaric, Condor MK3
3) OCT Standard [4]
4) Derived from the number of connections to an SDC

Table 9 depicts all primary (power, cost, weight) and secondary (inputs to other modules) results for a communication link, which are mathematically also expressed with subsequent block of equations. Default values are derived from default values of Table 8. They may vary strongly between different implementations/parameters of the model and from the system simulator. See also Section 10 for details about which equations are currently implemented in the simulation tool and which ones just serve for reference.

Table 9 Network Model Output Values (i.e. derived values) and Defaults (See footnotes for superscripts)
(blue rows for reference only = second priority)

Symbol	Description	Unit	Default Value	Eq.
n_{terms}	Number of physical links/terminals required	#	6	Eq. 8
dt_{link}	Expected link delay	µs	N.A.[1]	Eq. 9
R_{cache}	Probability of having to cache communication/network		0.1[2]	Eq. 10
P_{router}	Aggregated router power	Watt	12.5	Eq. 11
t_{link}	Approximate connection time for dynamic links	s	~ 1200 (20 minutes)	Eq. 12
t_{no_link}	Maximum time without any connection at all [3]	s	~ 1200 (20 minutes)	Eq. 13
M_{link_cache}	Minimum Cache Memory required	MB	TBD	Eq. 14

1) For current workloads likely not relevant and mainly governed by available router technology
2) Derived from Fig. 18 … 30 dg. Sun angle over 360 degrees = ~ 0.1
3) Caused by solar stray light blocking the optical link

$$n_{terms} = ceiling(\frac{tp_{link} \cdot 1/R_{cache}}{bw_{term}}) \quad \text{Eq. 8}$$

$$dt_{link} = \sum \overline{dt}_{\dots} = \sum f(\dots) \quad \text{Eq. 9}$$

$$R_{cache} = \frac{\alpha_s}{360°} \quad \text{Eq. 10}$$

$$P_{router} = bw_{tot} \cdot p_{router\_xx} \quad \text{Eq. 11}$$

$$t_{link} = d_{max} \cdot \frac{\sqrt{3}}{2} / v_{orbit} \quad \text{Eq. 12}$$

$$t_{no\_link} = \frac{S_{orbit}}{v_{orbit}} \cdot \frac{\alpha_s}{360°} \quad \text{Eq. 13}$$

$$M_{link\_cache} = t_{no\_link} \cdot tp_{link} \quad \text{Eq. 14}$$

Above equations clearly show how individual elements of the SDC network can be modeled. Discussions with experts in the field made it clear, that performance-wise the network is important, however, that it impacts power- and cost of an entire HPC or public cloud data center with less than ~ 10%. Because the 3 pre-selected SDC 3 use-cases are not expected to be network-limited (see next sections), we took the liberty to simplify the implementation of the network model in the simulation tool to the overall number of routers/switches required and their power- and cost contribution to the overall system.

6.3. USE CASE 1 – MOTHERSHIP AND SCOUT SATELLITE

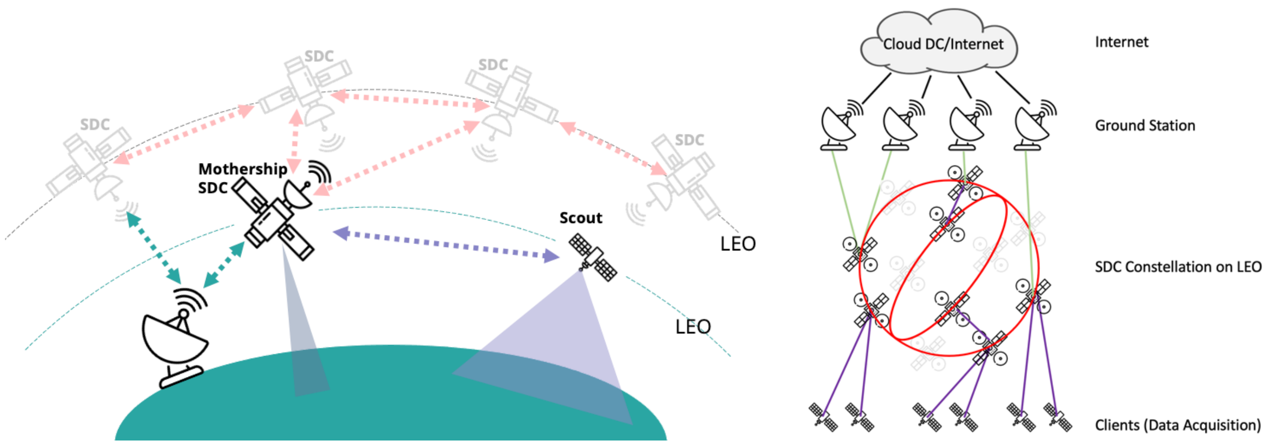


Figure 25 Mothership – Scout Network Architecture. Minimum scenario feeds directly from the mothership to the ground station. More generic scenario connects from the mothership to the SDC constellation to the ground station.

Let us consider how the system described above operates in the case of a mothership with a scout. The mothership can be considered a regular SDC with additional observation capabilities and an additional co-orbiting scout satellite. The logical connection between the ground station and the scout happens through multiple levels in the topology. Depending on the priority of the workload, many SDCs may be involved to provide immediate access to a ground station. The minimum implementation however, potentially with network traffic caching, involves:

1. A quasi-static router, on the scout and on the mothership SDC, as they are on the same orbit.
2. A dynamic router from the mothership to the ground station as soon as in reach

If either some of the compute is distributed or other resources of the SDC constellation is required, data may be sent to SDCs on another orbit first, which would then yield the following more generic scenario:

1. A quasi-static router, on the scout and on the mothership SDC, as they are on the same orbit.
2. A dynamic router from the mothership to one of the SDCs on another orbit
3. n quasi static routers between SDCs on the same orbit. n depends on the number of SDCs per orbit and the ground station density … assuming no caching is desired.
4. A dynamic router from one of the latter SDCs to the ground station as soon as in reach

Having these schemes allows routing up and down between clients and ground stations in an efficient manner. Using the ad-hoc mesh network, the external network router forwards the information towards the router to which it is currently connected. If no such connectivity exists, it waits till the connectivity is established. This provides disruption tolerant networking.

When disconnected, the locally cached content is used to serve responses as needed. Furthermore, data compression techniques are used to reduce the amount of data to be transferred during the connected phase. This allows a faster communication when the network is connected. Customized network services cache content and allow seamless operation when disconnected.

When connected, byte-caching and fast transfer techniques maximize the throughput possible.

6.4. USE CASE 2 – LEO-GEO OBSERVATION ARCHITECTURE

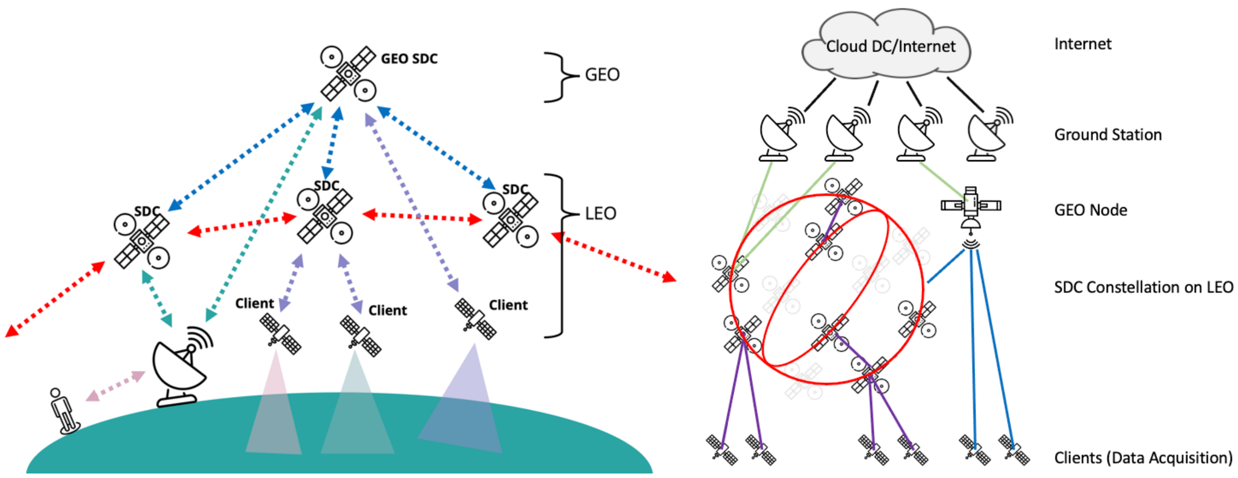


Figure 26 Additional GEO SDC for Uninterrupted Connectivity to Ground Station & Related Network Architecture (Network links in violet, blue and green with dynamic routers, links in red [on same orbit] with pseudo static routers)

In the original LEO-GEO Observation architecture, the SDC, with (router) communication capabilities, will be installed on a GEO orbit. A more generic routing scenario with additional SDCs on LEO orbit looks as follows:

1. A dynamic router on the client connects to a corresponding dynamic router on a constellation SDC

2. n pseudo-static routers propagate the date to a constellation SDC with good visibility to the SDC on GEO

3. A dynamic router on the latter constellation SDC connects with a dynamic router on the GEO SDC

4. A very long-range pseudo static router of the GEO SDC connects to the corresponding router on the ground station.

Apart from the addition of the optional GEO node, the architecture and approach are identical to the scout-mothership approach. We consider this the most generic approach, highlighting the flexibility of the overall SDC concept.

6.5. USE CASE 3 – LUNAR LANDER

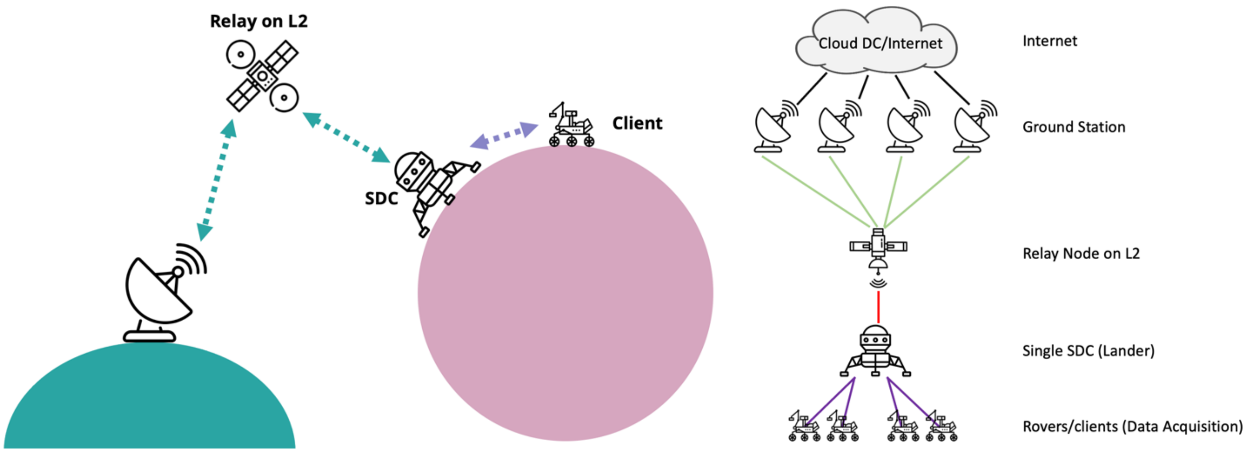


Figure 27 Lunar Lander Use-Case & Related Network Architecture

Also, for this scenario, the network architecture is similar to that of the previous use cases:

1. A pseudo-static router connects the (likely RF connected) rover (clients) to the lander SDC.

2. Using more "traditional" RF based connections, the SDC will communicate with a relay satellite on an orbit around L2 (Earth-Moon Lagrange point). As visibility from an L2 orbit to Moon and earth is always possible, also a pseudo-static router scheme can be implemented here.

3. The relay node will then directly communicate with ground stations on earth, which can likely provide a better receive quality than connecting to an intermediate LEO constellation. Given the rotation of the earth, a dynamic router scheme between L2 relay and ground station is necessary.

6.6. NETWORK ARCHITECTURE SUMMARY

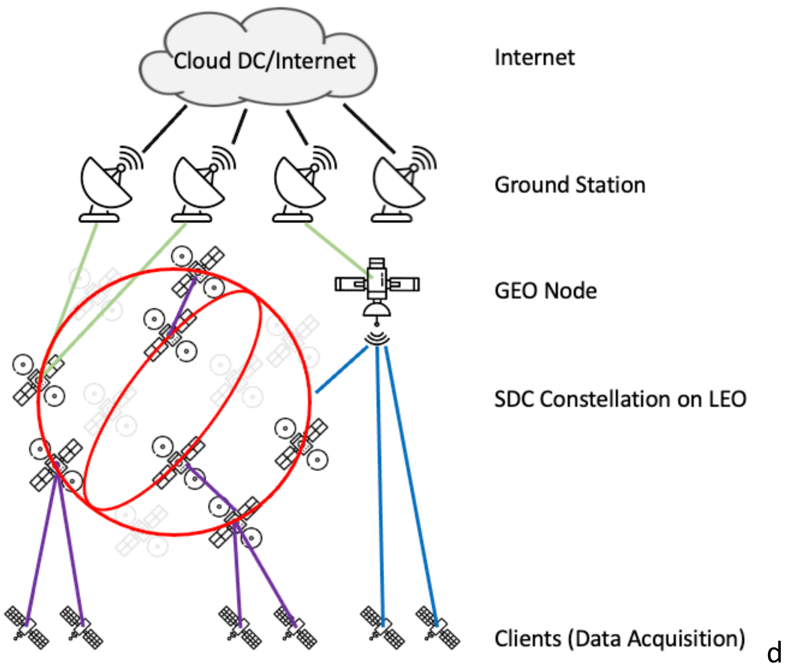


Figure 28 Generic Network Architecture, e.g., for Earth Observation Use-Cases

The generalized architecture of a network element in this architecture can be seen in Figure 27. In general, a network router is taking network packets from its downlink nodes that are closer to the terrestrial Internet and forwarding them to uplink nodes which are closer to the clients in space and forwarding packets from the uplink nodes to downlink nodes (or local space data center). As the network changes dynamically, the identity of the uplink and downlink nodes may change, but the forwarding behavior remains the same. The different space routers may be connected in a different hierarchical relationship as shown in Figure 27.

7. WORKLOAD ANALYSIS (WP5)

Authored by: Jonas Weiss

In this section, the workloads for the 3 preselected use-cases are studied and quantified. With additional input/dependencies from system requirements and system-architecture boundary conditions, the aim is to estimate compute, bandwidth/connectivity, memory, and storage requirements as a function of the workload to be executed.

Unless provided by explicit requirements, educated assumptions and state-of-the-art approaches from literature are assumed, e.g., when estimating algorithmic workload for wildfire detection as required for use case 1.

The goal is to map the outcome of this section into executable model-components of the system simulator (MS Excel VBA). For this purpose, relations and interdependencies are ideally functional relations or options thereof (if-then statements). As an example: The overall computational cost (Gflop/TFlop) for detecting wildfires is estimated to be ***X***, which will be distribute to ***n*** SDC nodes. This yields a node compute requirement of ***X/n + oh(n)***, where ***oh(n)*** constitutes the compute overhead caused by partitioning the problem to multiple nodes.

7.1. WORKLOAD DEFINITION

A compute workload refers to the amount and type of computational tasks or processing requirements that need to be performed by the system's hardware and software components. It represents the specific set of operations and calculations that the system must execute to fulfill its intended functionality.

Therefore, understanding and managing the compute workload is crucial to ensure the efficient and effective operation of a system.

A compute workload can vary widely depending on the specific application and requirements. In our case, the predominant task is sensor data processing to extract information from raw data and/or perform data-compression for efficient communication and temporary storage.

Additionally, feature extraction in the context of multimodal AI models to create more complex insights, involving data from different clients (satellites) and data-flow optimization for the latter can also be considered as relevant processing tasks.

The compute workload encompasses real-time processing, where tasks must be completed within specific time constraints to ensure proper system behavior and off-line (post processing) of data on storage to optimize the latter and fully exploit available compute resources. It also involves handling multiple concurrent tasks and managing periodic tasks with different priorities.

Efficiently managing the compute workload involves optimizing resource usage, balancing processing demands, and ensuring timely execution of critical tasks. This requires techniques such as task scheduling, power management, memory management, and algorithmic optimizations tailored to the system's constraints and performance goals.

7.2. USE CASE 1 – MOTHERSHIP & SCOUT (WILDFIRES)

In this use-case there are two data-sources involved: 1) The high field of view, but rather coarse resolution sensor of the scout and 2) the high-resolution, narrow-field of view instrument of the mothership. We assume that both workloads are being computed on the mothership, i.e. that scout raw data reaches the mothership over a dedicated (RF or rather optical) data link.

As a reference, we assume a 290 km swath width, like Sentinel 2, and an 800 km LEO orbit with ~ 7.5 km/s "orbital speed". This translates into a 290km/7.5km/s = 40 seconds to acquire a 290 x 290 km^2 image. For a coarse detection of wildfires, we assume 100m resolution and for the high-resolution image a 10m resolution with a 29 x 29 km^2 image (average US wildfire is ~ 0.5 km^2).

We thus have two workloads with identical input size:

- **Workload Data-Input:**
  2900 x 2900 x 3 pixel (8Bit) per 40 seconds (3 most relevant bands only) = 25.2 MByte/40 s
- Equals 460 x 460 x 3 pixel per second = 630 kByte/s

From Section 4.1.2 of the System Level Requirements Report, the computational cost of a "plain" U-Net, suitable for semantic segmentation and wildfire detection is reported to be

~1.66 million MACs (~ FLOPs) time Height x Width of the RGB (3 channels!) image, which can be translated into the following compute-per-data-volume figure:

- ~ 1.66 MFLOPs x 1000 x 1000 (/ 3 channels) = 550 GFLOP/MByte

This translates into an approximate compute requirement of ~ 550 x 0.63 = 347 GFLOPS. This number accounts for either the coarse or the fine resolution data.

In [12] we find the following table with compute effort as required for typical semantic segmentation tasks (building detection in this particular case), which serve as a reference.

Model	FLOPs(G)	Parameters(M)
FCN-8s	73.49	134.27
U-Net	16.59	31.06
SegNet	79.89	39.87
Deeplab v3	121.06	60.99
ENRU-Net	51.87	73.71

Figure 29 Semantic Segmentation number of Gigaflops per 256x256 RGB image.

This yields a range of GFLOPS[Table] /(256x256) x (1000 x 1000) /3 bands ≅ **90 – 600 GFLOP/MByte** for input images.

As the occurrence of wildfires is expected to be relatively small, we only consider the continuous coarse wildfire scanning only, to derive the average compute requirement.

7.3. USE CASE 2 – LEO & GEO OBSERVATIONS

For this use-case, there are no application specific compute requirements defined. To get a reasonable feeling for potential applications, we assume a similar workload as for use-case 1, but accounting for potentially multiple clients, i.e. the same compute complexity per data, but multiple times the data is assumed:

- N x 460 x 460 x3 pixel per second, with N the number of concurrent clients, thus N x 21 MByte of data per second that need to be processed
- This results into N x 347 GFLOPS of compute requirements

Due to the very long communication distances, power consumption for data-transfer will constitute a larger fraction of the overall power budget, compared to LEO-only scenarios.

Similar workloads, which like wildfire require "real-time" (low latency) situational awareness, are e.g. oil spills, flood-detection, vessel tracking, etc.

7.4. USE CASE 3 – LUNAR LANDER

From the rovers of a lunar mission, we expect the following:

1) The size of an average data-packet sent to the lander SDC is significantly smaller than for earth observation satellites. We assume an equivalent 1000 x 1000 x 3 pixel image at the time.

2) The framerate of terrain and landscape probing rovers is expected to be much lower than for other space applications too. Given the small moving speed of such vehicles, we assume a 60 second interval.

3) Initial number of rovers however, maybe "significant", compared to client satellites. We assume a value of 20 here.

For the lack of better data, we assume yet the same workload complexity as for the other use-cases, i.e. 550 GFLOP/MByte of received data at the lander SDC. This yields:

- N x 3MByte / 60s x 550 GFLOP/MByte = N x 27 GFLOPs

8. SOFTWARE STACK (WP5)

Authored by: Martin Schmatz

8.1. SOFTWARE STACK DEFINITION

8.1.1. COMMUNICATION BOUNDARY CONDITIONS OVERVIEW

Before any SW-stack can be defined, a few boundary conditions with respect to network communication principles must be defined because those will have a profound impact on the feasibility of distributed deployment SW-stack deployments.

Earth-based communication:

It is safe to assume that any compute-infrastructure deployed on earth can have network connectivity to all other earth-deployed compute-infrastructure at any time. Traditional Internet routing will solve the inter-cluster communication and communication to ground-stations, while standard netfilter-optimized communication will take care of any intra-cluster communication. Communication to off-cluster compute-acceleration nodes will leverage APIs over Internet.

Space-based communication:

There are two fundamentally different scenarios for space-based communication, best characterized as either active or passive: In the 'active' scenario, each space-based object acting as data or control source can enforce a communication to a communication sink and push data & control to that communication sink, while in the 'passive' scenario, a communication sink will have to poll a communication source to receive any data or control and a communication source will have to wait to be contacted by a communication sink before any data or control can be transferred.

In contrast to earth-based communication where the network endpoints are well known and stable, the space-based communication channels will move in both time and space. Consequently, the establishment of a communication channel also requires space and time knowledge of the location of the peer. Remark: Wide-field scanning could in principle be applied to gather such information in-situ, but as the orbits of space vehicles are well known, it is much less effort to establish a communication channel when this information is used.

This raises the question: “Which entity should know the orbits of space vehicles?”. It could be the earth-based clusters via the ground-station, the SDCs, or the scouts. Given the fact that (a) the connectivity from ground-station to SDCs is fluctuating, and (b) the scouts should be as light-weight as possible, it can be deterministically concluded that the optimum place to keep orbit information of all space-based vehicles are the SDCs.

When combining this insight with the ‘active/passive’ considerations from above, the SDCs are the ones which will initiate any communication, be it to the scouts, to the ground-station, or to other SDCs. The latter implies that SDCs must be dual-role capable: They must be capable to actively initiate communication, but also have provisions to passively accept communication connections.

Overall, this setup has many advantages:

- Scouts can be purely passive and hence don’t have to initiate communication, and therefore don’t need to maintain knowledge of orbit data; their communication subsystem can largely sleep during most of the time, only to wake up when contacted to forward their data to the SDCs;
- SDCs can decide whether or not to establish a connection with any of the scouts, therefore only consuming power when there’s a need for communication.
- SDCs can decide whether or not to establish a connection with any of the (few) ground stations, therefore ensuring that the available BW to/from ground stations is predominately used for precious data communication when data is actually available;
- Given the dual role of ‘active/passive’ of SDCs, communication between SDCs will consist of two channels, each established by one of the participating SDCs, greatly improving reliability via redundancy and doubling communication bandwidth.
- The ground-station still can – in case of emergency – connect to any space-based vehicle by establishing a direct communication path to those.

- Network setup (via communication establishment) is entirely handled in-space, freeing the earth-based clusters of that task, which also implies that space-based network setup is appropriately done even when there isn't any connection to the ground-station.

With the above, the SDCs become the main entities of network management in space, which must be appropriately reflected in the network stack.

8.1.2. SOFTWARE STACK OVERVIEW

The software (SW) stack and the related deployment principles must be *defined individually for all locations*, from Enterprise DCs all the way to the scouts.

Enterprise DC & (Public) Cloud:

OpenShift clusters with optional specialized number crunching compute facilities, in particular for AI acceleration, connected with APIs.

Ground-Stations:

Small, custom designed SW-components communication routing, offering APIs to both SDCs and Cloud clusters.

SDCs:

Phase P1: Each SDC will operate its own OpenShift cluster. Besides the functional data services, there will be dedicated services deployed to control in-space communication to either SDC peers or scouts.

Phase P2: Operation of a scaled down OpenShift cluster, plus participating in a space-wide OpenShift cluster with dynamic node appearance.

Scouts: Dedicated scouting SW, plus APIs which (a) enable acceptance of communication requests (from SDCs or ground stations), (b) serve data pull requests, (c) enable download of configuration and control.

Scout space-vehicles:

Specialized SW will be deployed on the scout space vehicles, which essentially will be sensor SW, tailored to the needs of the sensor capabilities, and enablement of temporary storage of acquired data. In addition, communication and data access SW with APIs will be deployed, which will enable SDCs to actively poll available data from the scout space-vehicles and download control data.

Network stack:

Besides the individual SW components for cluster management and services, it is worth noting that the network stack must leverage quantum safe communication capabilities and mTLS (mutual transport layer security). More details can be found in the relevant section 7.3.

8.2. DEPLOYMENT AND APPLICATION LIFECYCLE MANAGEMENT.

8.2.1. DEPLOYMENT PHASES OVERVIEW

Two distinct phases must be considered: (P1) Initial sparse deployment of SDCs and scouts, (P2) full deployment of SDCs. The two phases P1 & P2 differ significantly in the ability to form meshes of nodes in space.

Consequently, each SDC will be treated as a self-contained cluster in phase P1, while the entirety of SDCs in phase P2 will be treated as a single cluster with some dynamically added/removed nodes. During transition, each SDC will maintain its local cluster and gradually participate in the global cluster.

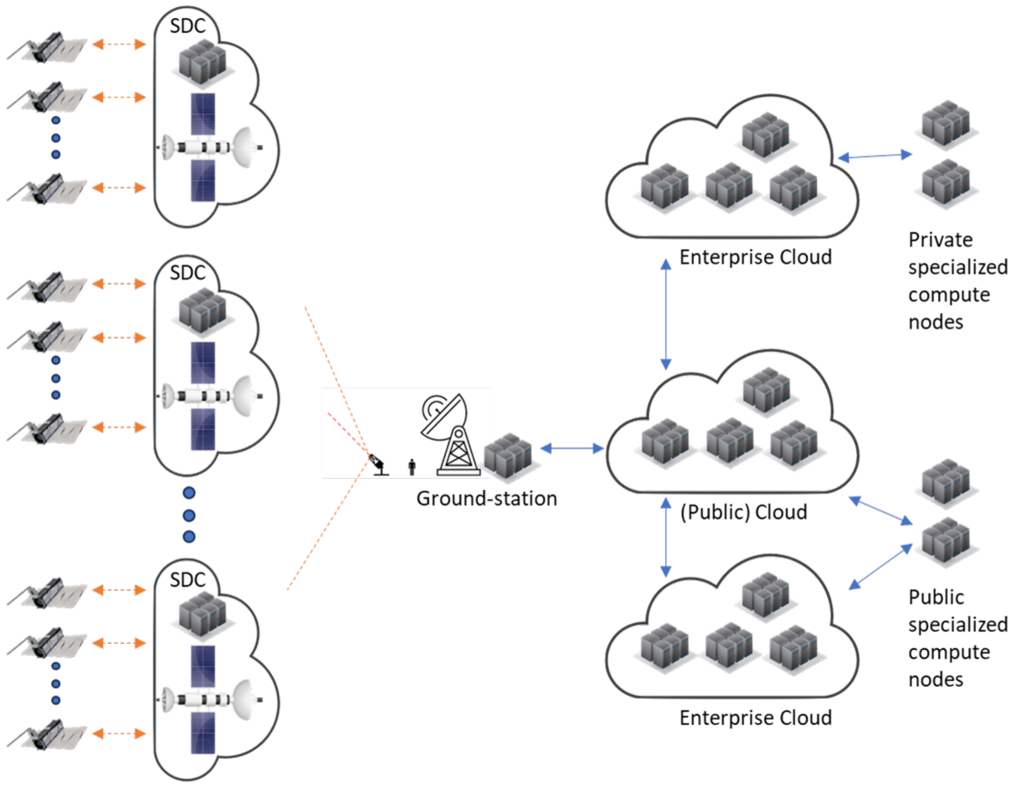


Figure 30 Phase 1 Self Contained Clusters

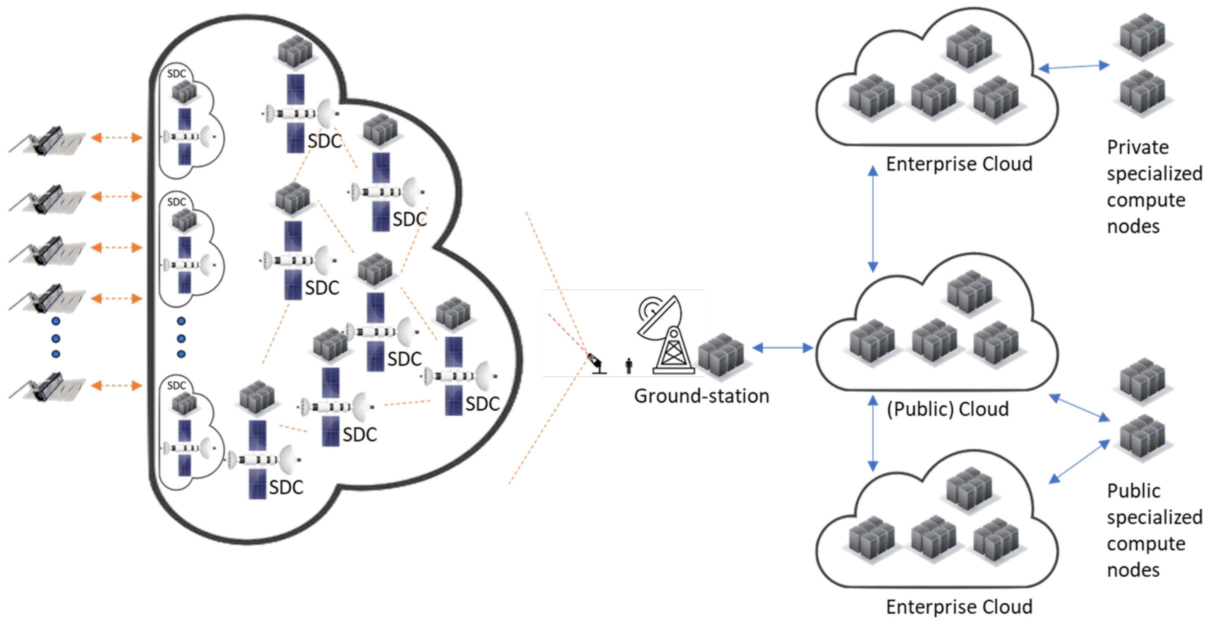


Figure 31 Architecture overview during deployment phase P2:
SDCs form a space-wide cluster with dynamic node appearance,
plus (logically) maintain per-SDC clusters.

8.2.2. APPLICATION LIFECYCLE MANAGEMENT

Enterprise DC, (Public) Cloud, Specialized compute nodes and ground stations:

The application lifecycle management for any earth-based deployment will follow state of art practices as applied in ‘normal’ SW deployments. That is well understood and does not require any further explanation in this document.

SDCs:

SDCs can establish communication channels to ground stations, thereby announcing their availability for any updates. At the same time, they offer APIs to upload any control data, which can be

- Cluster configuration
- Application deployment & revocation
- Application updates
- Upload of application deployment, revocation & updates for scout space-vehicles

In the initial deployment phase P1, each SDC is treated as an individual cluster, and standard cluster management SW can be leveraged. Once sufficient SDCs are deployed, the entirety of all SDCs will be treated as one single cluster, where again standard cluster management SW can be leveraged.

As in earth-based clusters, it is worth noting that such cluster management can largely be done without any service disruptions, clearly showing the advantage in terms of specialized management SW development (actually the lack thereof) when treating all SDCs as being a cluster of being part of a cluster.

Scout space-vehicles:

Scout space-vehicles operate in “passive” communication mode. This implies that an SDC will initiate any required SW lifecycle action (deployment/revocation/update) by establishing a communication channel to a scout space-vehicle whenever the SDC has received any such action request from the earth-based control entity (the cloud-based cluster). The SDC uses APIs offered by the scout space-vehicle to execute deployment/revocation/update actions on the scout space-vehicle.

8.3. BASIC SERVICES & SECURITY

8.3.1. BASIC SERVICES

Enterprise DC & (Public) Cloud:

The SW deployment will be characterized by consisting of OpenShift clusters, with optional off-cluster compute-nodes for special number crunching. Any compute-resources at the ground stations will be limited to communication tasks to/from SDCs.

The services running on the cluster can be categorized into:

(a) Cluster management.

(b) System management, including space vehicle management.

(c) Application services handing the application scenarios.

While cluster management can be off-the-shelf OpenShift SW, and the application services will be determined by the use case, the system management services will have to be custom designed to handle the ever-changing constellation of connected SDCs. During phase P1, the system management will treat each SDC as a single, external cluster. During phase P2, the system management will control (via APIs) the dynamically sized space cluster consisting of the entirety of SDC.

Namespace insulation will ensure that none of the three types of cluster services will influence the others.

In addition, the Cloud-based cluster must offer PKI (Public Key Infrastructure) functionality, with details found later in this section.

Ground-Stations:

Besides fundamental orbit observations (as envisioned to be required by the space vehicle providing company), the ground-stations functionality will be limited to pure network nodes. In other words, all the space vehicle control will be performed in the Cloud-deployed clusters.

The SW at the ground-stations will therefore be small, custom designed SW-components for network discovery and routing.

SDCs:

The SDC-based cluster(s) will offer three fundamental classes of services:

- Space vehicle management (SDCs and scout space-vehicles): This essentially establishes the control path for the SDCs. It offers APIs for cluster management, upload/updating orbit data of all space-vehicles, upload of any scout space-vehicle control data.
- Data processing applications (including data downloading to earth ground-station): This establishes the main data-path for the SDCs. It mainly consists of cron-jobs, which - according to control data received from the earth-based control station – will pull the raw data from the scout, apply preprocessing and/or data-fusion algorithms to that data, and finally will offer capabilities to download results to the earth-based cluster(s). It is worth noting that this enables to signal “data ready” to earth-based cluster(s), *without* actively downloading any data. The actual data download is then initiated by earth-based SW, which is free to use APIs to get ‘only’ preprocessed data or use APIs to get raw data.
- Data-processing and/or data communication offload services to other SDCs (applicable only in phase P2 when sufficient SDCs will be available): Given that SDCs are part of a space-wide cluster, workloads will automatically be distributed to nodes in SDCs which have spare capacity.
-

8.3.2. SECURITY

Secure Boot and SW Attestation:

First line of defense: To avoid injection of any rogue SW, secure boot principles will have to be ubiquitously enforced. In addition, and to ensure at any point in time that deployed SW, be it on earth or in space, be it cluster control SW or application SW, is authentic, standard SW attestation principles with remote attestation capabilities will have to be deployed. To enable this, installation of a hardware TPM (trusted platform module) or HSM (hardware security module) is a must.

Last-not-least, all deployed SW must be scanned for existence of any rogue code, like viruses, worms, side-channel-attacks and the like.

Communication:

Second line of defense: All communication channels must implement quantum-hard, authenticated protocols (e.g. QSC mTLS). Consequently, a QSC (Quantum Safe Cryptography) PKI (Public Key Infrastructure) must be an inherent part of the earth-based control cluster. Any space vehicle must be able to securely boot and generate private keys and send CSRs (Certificate signature requests) to the PKI *before* being launched. Certificates generated and loaded to the space vehicles in this way will be a first line of defense to any rogue network attack by leveraging the capability for ubiquitously using mTLS (mutual TLS) in any communication: It is guaranteed that only a-priory approved communication endpoints can establish a communication channel due to mutual authentication.

Provisions for in-space certificate renewal/rotation must be present. In addition, it is strongly recommended that each space vehicle includes a secret, at least 4k Bit long personalization number, preferably stored in hardware on the space vehicle and in a HSM on earth. This can be used as emergency fallback to derive authentication or encryption keys, should any security breach be detected or suspected.

IAM (Identity and access management):

Third line of defense: The next line of defense will be implemented via IAM (identity and access management). While standard practices can be deployed for all earth-based equipment, the deployment of IAM on space-based vehicles is anything than trivial. The reason for this is the lack of access to a centralized IAM entity. To circumvent this issue, each SDC will operate its own IAM sub-system, which is configured in two steps: An initial, unalterable setup will be done before launch. This opens the capability to implement workload insulation on a coarse granularity: Any initial (and paying) rider will get access to deploy workloads with full insulation from other workloads. Any entity deciding to leverage SDCs at a later stage will have to rely (and trust) the IAM setup done by the control entity while SDCs are already in space.

Cluster administrative access:

Fourth line of defense: The access to any administrative function of the cluster must be limited to a small number of administrators, with dual factor authentication.

Production supervision:

Automated tools (e.g. CrowdStrike or similar) should be deployed to constantly monitor all deployed systems (except for the scout space vehicles). This to detect any rogue and even more importantly any vulnerable software and flag it for update(s).

8.4. PROPOSED STACK OPTIONS

Cluster type:

OpenShift is highly optimized for a secure enterprise environment. It also inherently features observability and logging functionality, and it is supported by all major cloud providers. OpenShift provides security response teams, long-term support options, validated third party operators, certified databases and middleware, and meets requirements for large-scale operations. The entirety of these features made it the first choice to be used as cluster SW by quite some margin.

The open-source version of OpenShift, named OKD, is a community project of packaged software components needed to run Kubernetes. OKD is the upstream project of Red Hat OpenShift, optimized for continuous application development and deployment. It provides the same functionality as OpenShift but requires more installation effort and comes without any support (other than from the community).

Native, open-source Kubernetes is the basis of OpenShift. Should the additional security and/or functionality of OpenShift not be necessary, this cluster type could also be an option. Like with OKD, it would come at higher installation and maintenance cost, and without support (other than from the community).

Operation mode:

The proposed cluster structure by design is for *multi-tenant* operation. While it is preferrable to operate one single cluster for cost optimization, it is technically feasible to operate independent clusters in parallel. Such clusters could vary in size and hence compute capabilities.

The advantage of operating more than one or even several clusters in parallel is that *single-tenant* operation modes can be enabled. One premium user can operate its own cluster, ensuring the even the SDC owner would not have any access to neither algorithms nor data anywhere. This enables the *secure compute mode* of operation for a premium user.

The SDC operator would only deploy (an attested version of) the cluster, but then hand-off ownership to the premium user.

It is worth noting that this can also be done dynamically: A user could temporarily rent single-tenant cluster capabilities to exclusively preprocess data and/or perform data fusion using his proprietary algorithms and data. Once the data processing is completed (and downloaded to earth-based DCs), the single-tenant cluster resources would be handed back to the SDC operator.

In addition to dynamically taking over a single-tenant cluster, it is also technically feasible to grow and shrink cluster size dynamically: A premium user can *temporarily rent more compute capabilities* as situational required.

Overall, the proposed SW stack, which essentially is a replication of the setup of a Cloud provider, enables the same variety of operation modes as available from typical Cloud providers, with many technically feasible operation modes. The market will finally determine which of the operation mode is most cost (or revenue/profit) optimized.

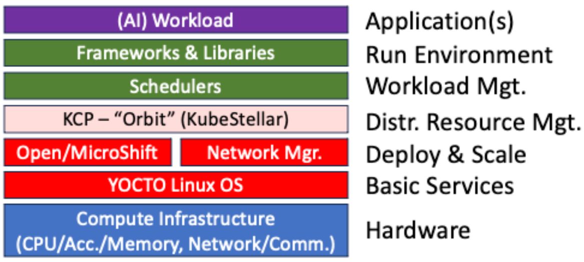


Figure 32 Generic Software (Hardware) Stack for an SDC Node

8.5. SIMULATION MODEL CONSIDERATIONS

A simulation model for the software stack needs to consider (a) the deployed infrastructure to determine cost, and (b) the workload to gain insights on energy consumption. The combination of the two aspects allows the simulator to draw conclusions about FOM (figure of merit) with respect to cost/energy.

Deployed infrastructure:

The variables used to characterize the deployed infrastructure must include (number and capacity):

- The compute capabilities (CPUs and accelerators like GPUs, tensor engines, AI accelerators, data compression engines and similar), for each location (SDC, Cloud, Enterprise, special compute nodes);
- The amount of installed memory in SDCs and related memory bandwidth, for each location (SDC, Cloud, Enterprise, special compute nodes);
- The network bandwidth for each class of connection: Scout space-vehicle to SDC, SDC to SDC, SDC to ground-station(s), ground-station to public/enterprise Cloud, and Cloud to special compute nodes;
- Earth-based short-, mid- and long-term data storage (Flash → hard-disks → tape);
- The number and data generation capabilities of scout space vehicles.

Workloads:

The variables used to characterize the workloads must include:

- The amount and frequency of data generated by the scout space-vehicles;
- The polling frequency from SDCs to scout space-vehicles;
- The data processing complexity of any data the SDCs receive from the scout space-vehicles;
- The actively used network bandwidth to transfer data from scout space-vehicles to the SDCs, from SDCs to SDCs (including any cluster management traffic), from/to SDCs to/from ground-stations, earth-based network traffic;

Simulation procedure:

Fundamentally, the simulation could start with a specific infrastructure deployment scenario, and then check cost/energy consumption for given workloads; or specific workloads could be defined, and the deployed infrastructure could be varied to gain insights on cost/energy.

The latter approach has the advantage that performance trade-offs can also readily be determined. Therefore, it is recommended to leverage a "fix workload → variable infrastructure" approach for simulations should be used.

For each workload, characterized by the variables defined above, the listed variables for the deployed infrastructure can be varied, and the result for cost/energy can be noted. Optimization procedures can be leveraged which would automatically tune the infrastructure deployment variables to achieve optimum values.

8.5.1. SIM COMPONENTS: OS

Provisioning of basic services by the operating system generates a certain amount of compute overhead. This overhead needs to be defined (estimated) for every hardware configuration and is independent of the algorithmic complexity of e.g. AI algorithms that run on the system.

Provisioning an exact value is first and foremost not generically possible, but also beyond the scope of this study. Consequently we will give a high-level estimation, which maybe refined either later during the project or by users with specific insights into their hardware and software stack.

For the sake of simplicity, we currently fully neglect the OS compute overhead, with the assumption that workload compute including accelerators is substantially beyond the

8.5.2. SIM COMPONENTS: OPEN/MICRO-SHIFT

OpenShift provides management services for deployment and scaling of Docker containers. It is fair to assume that the more containers will be deployed concurrently, the larger the compute load on OpenShift. For quantitative values, thorough application testing on dedicated hardware would be necessary, which is beyond the scope of this study.

9. SDC NODE (POD) ARCHITECTURE (WP7)

Authored by: Jonas Weiss, Martin Schmatz, Ingmar Meijer

In this section, we consider an SDC node to be a single satellite payload, i.e., NOT an assembly of spacecraft that have been launched sequentially and were (mechanically) connected in space.

With the goal to exploit mostly COTS components for the payload section and the resource constraints of typical satellites, we combine insights from large scale datacenters (architectures) with edge-computing considerations.

We focus on highlighting several design-options which are modeled in terms of cost, weight, and performance in a parametric way, such that “any” use-case specific system requirement can be quantified through the proposed simple parametric model.

9.1. HIGH-LEVEL PAYLOAD REQUIREMENT

The following high-level system key-requirements as summarized in Section 4 for the SDC payload-section can be identified:

1) Provide as much compute as possible, given a specific payload weight (driven by available power and cooling, thus power-efficiency of compute will be key)

2) Support state-of-the-art operating systems and software stacks to facilitate an open infrastructure ecosystem.

3) An entire satellite should fit into a weight envelope ranging from 150 kg (OneWeb/Starlink) to ~2000 kg,

4) and a power envelope ranging between 2-6 kW, which are typical values for smaller commercial satellites.

We further make the following (simplifying) assumptions:

a) A fixed ratio between the power of the satellite core functions (flight control and management) and power available for the payload, e.g., a ratio of 1:9

b) A fixed ratio between overall satellite weight and payload, e.g., a ratio of 1:2

9.2. STATE-OF-THE-ART PUBLIC CLOUD & HPC DATACENTERS

In Section 5 and 6 (terrestrial) data center architectures have already been addressed from an architectural and networking perspective. In this section, we'll have a closer look at how the inside of data center pods or racks are built, under what boundary conditions and how they can be applied to SDCs or where different considerations must be made.

9.2.1. PUBLIC CLOUD DATACENTER

"Classical" massive-scale-out data centers serve a large and diverse market, where, e.g., dynamically assigning resources for pay-as-you-go/scale models is a key capability. For example, webservers, webstores, some data-base applications, or moderate size data-insight applications must be allocated on compute, storage, and networking resources. Most of these workloads are comparatively small and will often fit into a single compute server, or even more likely, only require a fraction of the latter, i.e., can share a compute server with (many) other tasks. While there may still be significant traffic between different servers and racks within a public cloud datacenter, communication bandwidth and latency may, compared to HPC systems, be relaxed. Consequently, the physical data-transport distance between individual units is less critical, and cost and maintenance considerations are predominant design and technology driving factors. To relax requirements on power-delivery and cooling infrastructure (and to allow so-called free-cooling), generally less-dense servers are deployed in public cloud data centers. Figure 33 depicts a generic cloud data center architecture and a server chassis, optimized for total cost of ownership.

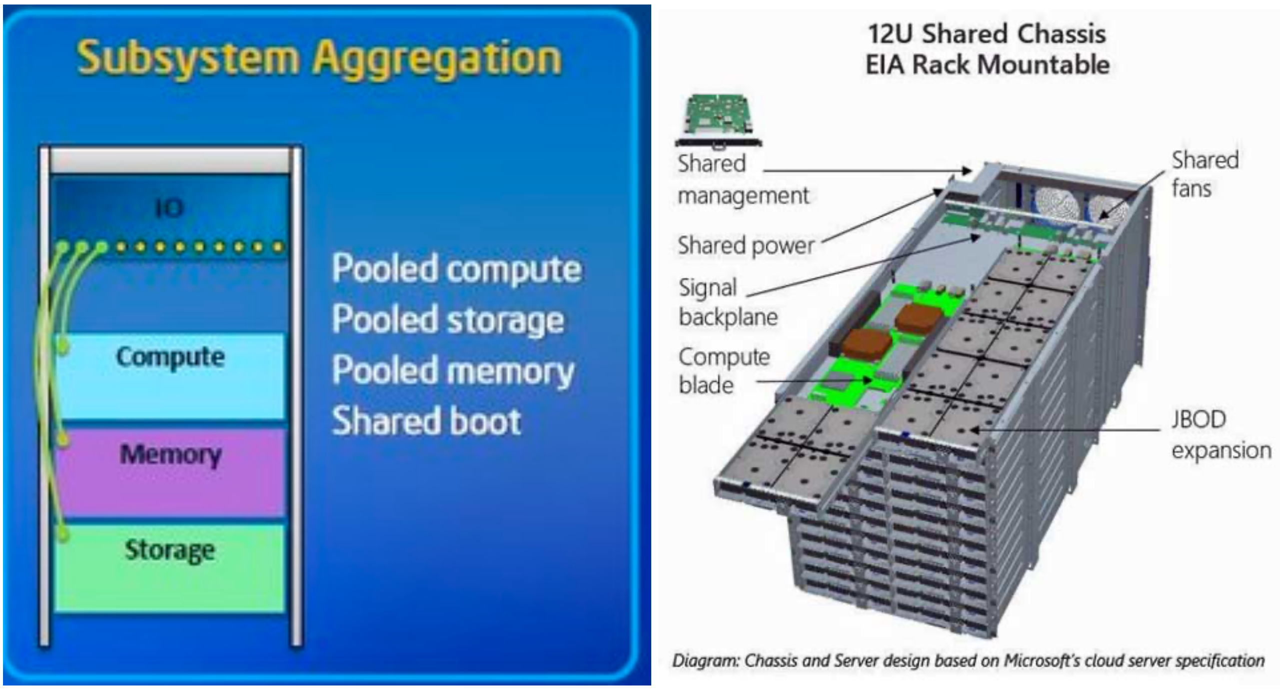


Figure 33 Generic Data Center Rack and Server Architecture

9.2.2. HPC DATA CENTERS

High-Performance-Computing workloads are typically very large and require orchestrated collaboration of hundreds to thousands of nodes. In large AI applications or physics-based simulations such as weather-, fluid dynamic-, or chemical process-simulations, the bottleneck is often not the compute capability of a single compute instance, but rather the data-exchange (e.g., boundary condition or other intermediate results) between individual instances. It is thus critical to maximize communication bandwidth and minimize communication latency between the participating nodes and maximize the compute performance of every single node, to, in turn, minimize the number of nodes required and thus latency induced by inter-node communication. For many HPC problems however, still a large number of nodes (or servers) is required, thus high bandwidth and low latency networks are always found in HPC systems. HPC systems generally impose severe constraints on power delivery and heat-removal.

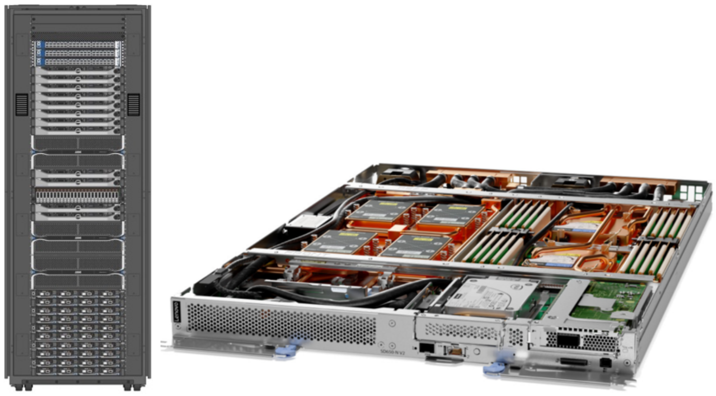


Figure 34 Generic HPC Rack and Server Blade – Notice the very densely packed components

9.3. EDGE COMPUTE (REFERENCE) SYSTEMS

In contrast to data center and HPC systems, edge-compute systems are deployed for different tasks but are also exposed to quite different constraints. While the definition of “edge” is vague and may range from IoT devices all the way to entire racks of compute servers in remote locations, they mostly have two things in common:

1. They mostly execute relatively narrow defined workloads and are not meant to provide generic and scalable capabilities comparable to datacenters.
2. They are resource constraint and driven by cost- and energy-efficiency considerations, sometimes also weight and volume, rather than by maximum possible performance.

Edge technology is found in systems such as (road) surveillance systems, data (traffic) management in cell-towers, embedded systems in production environments, environmental monitoring devices and systems, car-control and autonomous driving systems, cargo vessels, airplanes, in the oil- and gas-industry, renewable energy (production) and distribution and many more.

In terms of “architecture”, edge technology is similarly designed as other compute-hardware, just with a strong emphasis on above mentioned constraints. Figure 35 depicts exemplary applications, which mostly underline above observations of narrow workloads, remote installation, and limited resources.

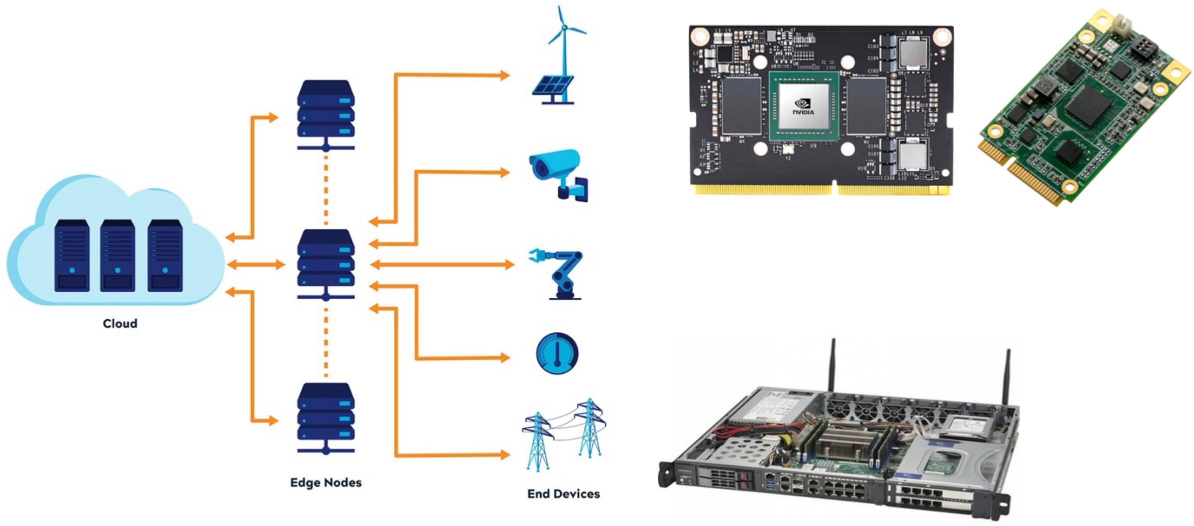


Figure 35 Edge Compute Scenario, Architecture, and Implementation Examples (Left, general edge architecture for different use cases, top-right: field deployable edge compute modules, bottom-right: edge-deployable server blade)

Often edge-compute technology is driven by, or can at least profit from the high-volume mobile market, which shares many of the encountered constraints, though typically is much more short-lived. Thus, depending on applications, the latter needs to be factored-in when making design-choices.

9.4. DC, HPC AND EDGE COMPUTING VS. SDC REQUIREMENTS

To better understand what technologies and design-considerations should be adopted for SDCs from public cloud DCs, HPC, and Edge-Compute, key aspects and requirements of all of them are compared in the following table. Clearly, edge-computing will be an important source for inspiration, while for certain aspects, also concepts from cloud datacenters, and even from HPC systems may be adopted.

An important learning from studying HPC- and scale-out cloud datacenters unveil that the power consumption of networking components is typically only 5% of the entire datacenter. This suggests that from an energy and efficiency perspective, the networking components and architectures leave little room for optimization.

Table 10 DC/HPC, Edge Computing and SDC Hardware Requirements Comparison (green = potential source of inspiration for SDSs)

Requirement	HPC Data Centers	Scale-Out Cloud Data Centers	Edge Computing	Space Data Centers
Compute Density	Maximize for shortest possible communication paths	Moderate: Cost driven	Low	As high as possible due to launch volume limitations
"CPU" performance	As high as possible to minimize number of nodes required per workload.	Moderate: Cost- and efficiency driven	Efficiency sweet spot to best exploit available power	Efficiency sweet spot to best exploit available power
Memory	Usually not too high	Rather high	Moderate	Moderate to high
Storage	Mostly large.	Separate "service" (large)	Small	Potentially very large.
Network & Communication	Ultra-high bandwidth and low latency	High bandwidth, less latency constraint	Comparatively low	Moderate, may be high for larger, distributed workloads
Power-Efficiency	Secondary, for operational and cost reason though	Important – it is an operational cost driver	High, due to edge power-constraint	Very high, due to space power-& cooling constraint
Weight and Volume	Weight given by floor capacity, minimize volume to reduce communication distances	Moderate – driven by real-estate/floor cost	Rather low, depending on specific constraints	Very low due to launch- mass and volume overall cost drivers
Reliability & Technology Availability	Moderate: Maintain over +/5 years, usually not mission critical	Moderate: Failing components may be replaced by similar, but not identical ones (5-10y.)	High, due to potentially limited access, staff availability	Highly redundant and graceful degradation. 3–5-year system lifetime
Cost	High (less sensitive), performance driven.	Low: Large scaling dominates overall cost	Rather low, but may be driven by redundancy and reliability requirements	Dominated by launch cost, thus rather weight, volume and thus also power limited (less sensitive)

9.5. GENERAL SATELLITE OUTLINE

This section is a complement to put the subsequent compute payload discussion into context. The discussion here is partly based on [5] .

Satellites are typically divided into a payload section and a (standard) bus section. The first is mission specific and in this case consists of the data center (compute, storage, networking, and communication) components, the latter consists of the supporting infrastructure, comparable to the building and other infrastructure of a terrestrial data center. Figure 36 depicts the different building blocks required, which are briefly discussed in the following.

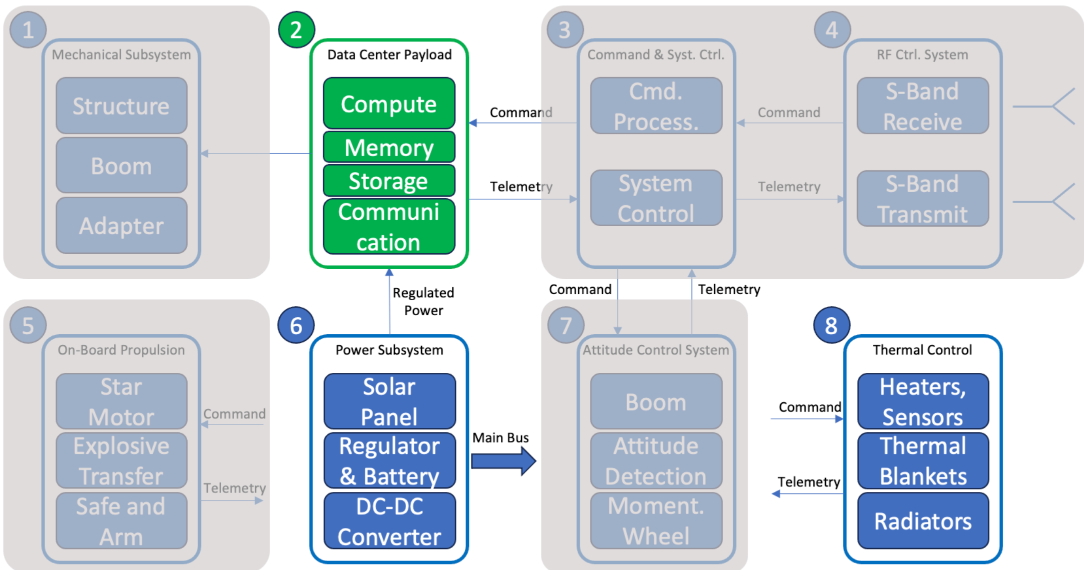


Figure 36 Generic (Small) Satellite or SDC Architecture.
Greyed out: Non-SDC specific bus-components to be considered at a consolidated level for this study.

9.5.1. SATELLITE BUS SUBSYSTEMS

The following are high-level considerations for the greyed-out building blocks is followed by a more details of the payload-, power- and thermal control subsystems.

1) Mechanical subsystem

a. Structure: Keeps payload, bus and all subsystems together
b. Boom: Maybe present to either keep instruments, antennas, or solar sails
c. Adapter: Facilitates attaching the satellite to the launch vehicle (rocket)

3) Command and System Control

a. Command Processor: Interprets and executes mission control parameters.
b. System Controller: Keeps the system running according to mission parameters.

4) **RF Control System**
    a. S-Band Receiver: As a fallback when (3) is not available through the SDC network.
    b. S-Band Transmitter: As a fallback when (3) is not available through the SDC network.

5) **On-Board Propulsion**
    a. Star, Motor: Navigation system and actual thrusters to maintain/change orbits.
    b. Explosive Transfer: Ignition and control of propulsion system
    c. Safe and Arm: Propulsion safety access system components

7) **Attitude Control System**
    a. Boom: Weights (some components) at the tip increase rotation inertia (stabilize)??
    b. Attitude Detection: Inertial navigation system and star/planet trackers
    c. Momentum Wheel(s): To maintain or adjust satellite orientation (e.g., to nadir)

For (3) and (4), we assume fixed cost, weight, and energy consumption per satellite, they are mostly independent of satellite size, per type of satellite. For the cost, weight, and power-consumption of (1), (5) and (7) we assume a linear relation to total satellite size.

9.5.2. SATELLITE PAYLOAD

The satellite payload (2), the in-space data-center, is assumed to constitute a “fixed” fraction of the over-all satellite weight, and equally, to consume a “fixed” ratio of the available power of the on-board systems. Both ratios will be user-selectable in the simulation-tool. Overall payload will be derived from available power, which in turn governs the satellite weight and resources.

9.5.3. SATELLITE POWER SUBSYSTEM

These components are responsible for generating (photovoltaic), converting (power-convertors), storing (batteries) and delivering (more convertors and cable harness) power in a suitable manner to all other system components. Individual components are governed by power availability, orbital parameters for buffering and payload requirements.

9.5.4. SATELLITE THERMAL CONTROL

Thermal control is achieved by radiation only. This limits implementation options to satellite internal thermal distribution-systems and availability of suitable radiators and their thermal feeding elements.

9.6. SDC REFERENCE COMPUTE ARCHITECTURE

9.6.1. ARCHITECTURE OVERVIEW

Given the limited power-envelope of SDCs,

Table 10 suggests that mostly edge compute hardware should be employed in the SDC payload section. While many of such devices combine compute and memory in a system on a chip (SOC) approach, some also don't. For this reason, the architecture as depicted in Figure 37 maintains all key element of a compute node as separate elements.

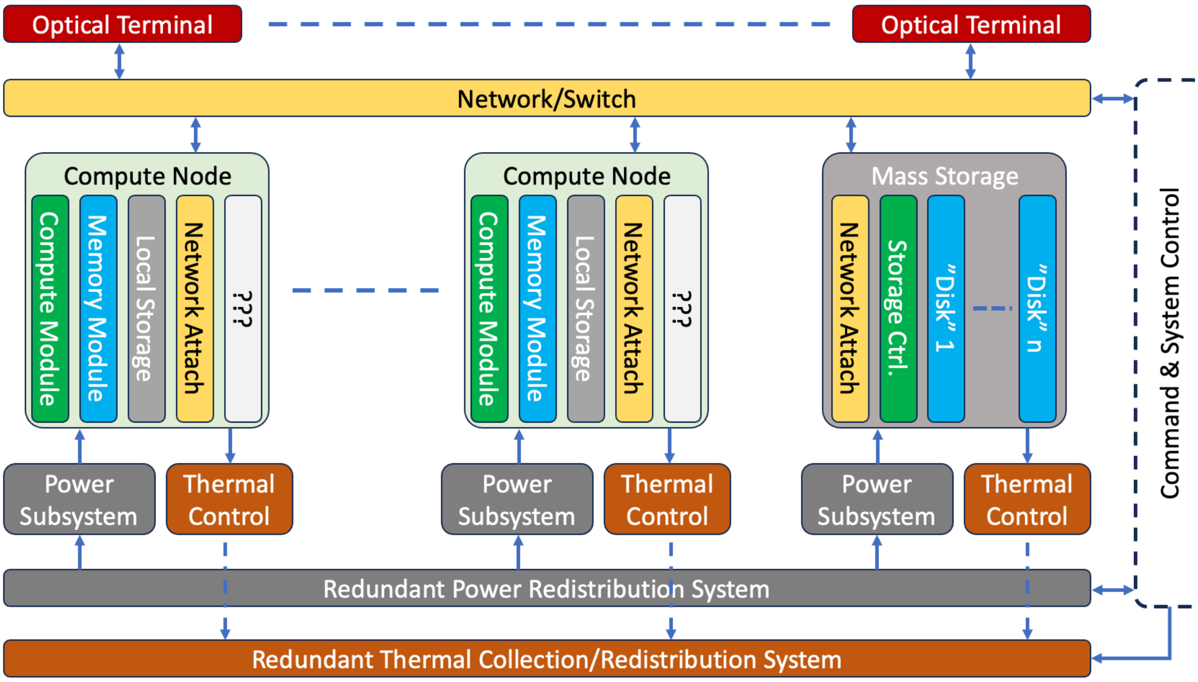


Figure 37 Single SDC Satellite Generic Compute Architecture.

In view of flexibility and redundancy, we suggest that each compute node has its own power subsystem. The latter can range from being a simple switched power convertor to transform the voltage from the solar panel (bus) to the voltages required on the compute node, as far as having a dedicated solar panel for every unit, making (excess) power available to other modules through the power redistribution bus, in case, e.g., aged or damage individual solar panels of other modules. This approach also simplifies quantitative assessment, as cost, weight, and other overhead scale linearly with the number of compute modules.

9.6.2. COMPUTE ARCHITECTURE & TECHNOLOGY

As was outlined above, the overall system cost is mostly governed by the launch cost of the system. In that view, as much compute as possible should be packed, in a lightweight and low volume compartment, while accounting for energy efficiency.

The following table gives a rough overview of different compute platforms that could be deployed for SDC applications – a summary from the Enabling Technology Report of Section **Error! Reference source not found.**

Table 11 Compute Architecture and Technology Comparison

Technology	Description	GFLOPs/W (2023)
CPU	Very flexible, programmatic compute unit. Specifically suited for complex operations that require context switching and branching in the computational graph. Does NOT provide a lot of parallelism for data-intensive workloads.	1.. 6[1]
GPU	We consider "edge GPUs", as initially no training of deep learning models is foreseen. With many data-pipelines, GPUs are optimal for data-intense, non-branching computational graphs. Note that also for GPU systems, a fraction (≥ ¼) of CPUs is needed.	~ 1000[4] (NVIDIA Jetson)
TPU	The Google Edge TPU is take as an accelerator specifically for AI acceleration.	~ 2000[3]
FPGA	While FPGAs can be very powerful devices, specifically in combination with embedded CPU (e.g. ARM) cores, they tend to be more power-hungry than application specific circuits. We lack exact numbers but anticipate ~ 5x improvement over GPUs in Training and 2-3x for inference.	~ 2500[2]
ASIC	ASICS can be "perfectly" mapped to workload requirements, thus very efficient, but with little flexibility for future workload- or algorithmic changes.	~6000

1. https://web.eece.maine.edu/~vweaver/group/green_machines.htm
2. https://www.run.ai/guides/gpu-deep-learning/fpga-for-deep-learning

3. https://coral.ai/docs/edgetpu/benchmarks/
4. https://developer.nvidia.com/embedded/jetson-modules

9.6.3. MEMORY

As was outlined in Section 8, it is highly recommended to overprovision memory, to the extent that volume and weight allow. The latter in view of that idle processes may reside in memory and not consume (significant) CPU (compute cycles), while higher priority tasks can exploit available compute.

3D stacking using COTS memory dies, e.g., with an additional redundancy and error correction controller [9] seems particularly promising for space applications for the following reasons:

1. Use of COTS components
2. High-density package, i.e., low volume and reduced weight
3. Radiation hardness through algorithms and redundant components
4. Reconfiguration flexibility in case of changing "space environmental conditions" or ageing effects.
5. Matching thermal footprint of compute "chips" allows to use similar or identical thermal attachment and distribution mechanisms.

Alternatively, to COTS memory devices, there are also technologies such as MRAM (as e.g., used in Copernicus Sentinel satellites), with intrinsically better radiation tolerance, but significant higher price and potentially lower performance than state-of-the-art COTS devices. From an architectural perspective, they can be treated similarly to COTS devices.

Connection to the compute unit is expected to occur through dedicated broad-band interfaces, i.e. following DDRx standards.

9.6.4. INTERNAL NETWORK

As outline schematically in Figure 37, we expect the individual modules per satellite to be connected in a peer-like fashion through a TCP/IP Gigabit Ethernet switch, i.e., NO hierarchical tree-like structure (Spine-Leaf) as is found in larger compute aggregations. On the scale of a single satellite, this allows simple scaling and use a single architecture to adjust a base-design to specific needs, e.g. trade compute modules vs. more storage or vice versa.

9.6.5. INTERFACES

While without a doubt there will be several “general purpose” interfaces to attach thermal-, power- and other SDC payload specific on-board sensors, we don’t consider them relevant enough at this level of a discussion to detail on them.

System management and interaction is expected to happen exclusively through commands over the main network (Figure 37) SDC Storage.

Data Storage on SDC is necessary for the following tasks:

1. Hot Storage, e.g., for operating system (OS) partition and operational purposes
2. Hot-Storage for long-term network traffic buffering (minutes or more of interrupted connectivity to the rest of the constellation).
3. Hot Storage for less prioritized applications data
4. Cold Storage for (compressed) raw data

As depicted Figure 37, we suggest using fully self-contained network attached storage modules, based on COTS technology. Using high-speed Gigabit Ethernet, this should be sufficient as well for hot-storage applications.

Regarding suitable technologies, SSDs are weight and power-wise an ideal storage technology, and without any moving parts, an ideal candidate for in-space applications. However, the used NAND flash-technology is known to be specifically susceptive to radiation. New (split cell) designs however [10] , seem to increasingly overcome this challenge and may constitute a viable option for the future to make weight, volume, and power advantages of flash-technology available to space-applications and SDCs.

9.7. SDC INTERNAL NETWORK

As depicted in Figure 37, we assume that long-range communication terminals will be directly connected to the network through TCP/IP Gigabit Ethernet or comparable technology. These terminals are point-to-point, and we expect them to have limited intelligence in a sense that if data cannot be transmitted/received and device-storage capacity has run out, the terminal itself will manage this and temporarily store data on the network attached hot-storage module, for later retrieval, or communicate with a constellation management service to elaborate on alternate data-paths for high-priority information.

The connection between clients and the SDC will be similarly managed by dedicated terminals. While technologically these terminals maybe different, e.g., optical downlink and RF uplink, potentially with one receiver for multiple clients, they will also be directly connected to the main SDC switch/router for easy integration into the overall network topology.

9.8. POWER SUBSYSTEM

A satellites power system consists of a primary and a secondary power system, where we assume solar panels/sails as primary and suitable batteries; traditionally NiCad, more recently Li-ion batteries as secondary power system, responsible for providing peak power requirements and power during solar eclipses.

A power failure on a satellite can have catastrophic consequences, as it may result in loss of attitude, thermal and other control and as has historically shown several times, to the total loss of the satellite. It is thus essential that:

1. The power system of the core functions of the satellite (satellite bus) is redundant and reliable.

2. The satellite bus power system is sufficiently decoupled from the payload section, such that payload (electrical) failure does not affect critical system power.

It may further be advisable to:

1. Have independent power delivery and management of subsections of the payload, to facilitate graceful degradation in case of partial power failure.
2. Physically separate individual power system to both electrically and thermally decouple them.

While we focus here on the power-system of the SDC payload, it may be tightly linked to the overall satellite power system, both in term of reliability and performance. To simplify things, we will calculate the overall satellite power by adding a fixed fraction of the payload power consumption.

Accounting for above constraints, we assume that every submodule of the SDC has its own power conversion and delivery module, depending on overall satellite geometry, it may even have its own, dedicated fraction of the total solar sail – but the latter is irrelevant for the following consideration. Besides operational benefits, this assumption also allows to apply simple scaling laws for modeling. Traditionally, high-quality, crystalline triple-junction solar cells have been used as a primary power system. They are expensive, relatively thick, and heavy, and susceptible to radiation. More recently developed rollable solar cells have much less electro-optical conversion efficiency (10% instead of 30%+), but potentially overall favorable power-to-weight ratios. Besides obvious operational advantages, they also offer more integration options, e.g., distributing individual components along larger surfaces, which in turn opens more options for heat removal and thermal management.

9.9. THERMAL CONTROL

Thermal control of all components with the satellite are of great importance. There are components such as batteries, or sensors that may require heating during specifically cold periods of the mission/orbit, while many electronic components will require cooling during operation. Very specific situations may also require compute circuits to be heated, e.g., for cold starts.

Regarding the SDC compute infrastructure payload while operating, key will mostly be local heat removal, distribution, and radiation.

Figure 38 depicts a concept of a relatively flat satellite, thus with a large surface-to-volume ratio, that uses the side exposed to the sun for energy collection and the rear facing side to get rid of excess heat by radiating to space.

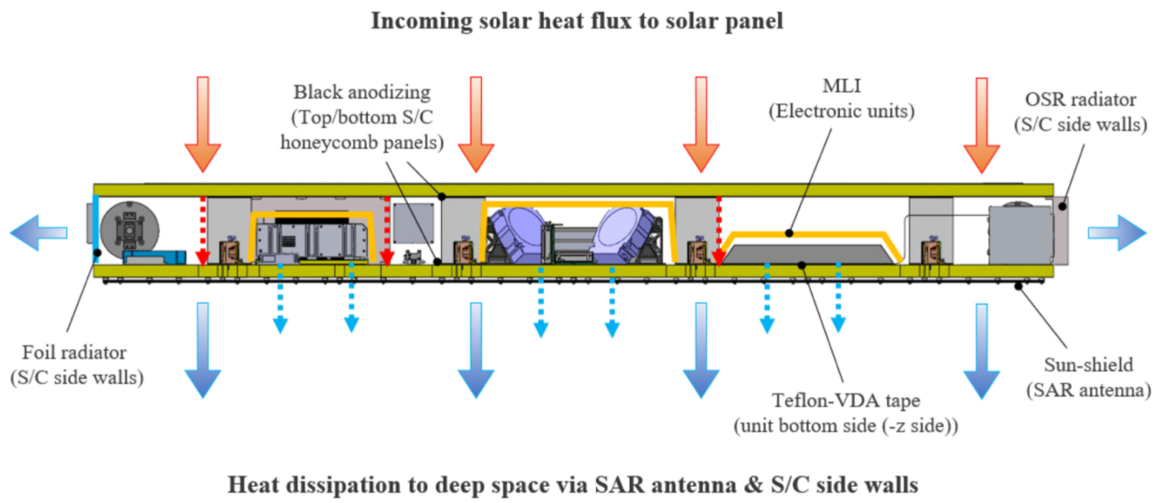


Figure 38 Satellite Thermal Concept – Deep-Space side for Radiation

Accounting for all options for thermal design and management is beyond the scope of this document. For modeling purposes, black body radiation of the power consumed by the SDC compute infrastructure is assumed and mapped to a corresponding surface area, which can be translated into weight and cost with some arbitrary, but in the future refinable values.

9.10. SCALING, COST AND ENERGY OPTIMIZATION

Scaling:

There are three aspects to be considered related to scaling:

- *System scaling*: The overall system can inherently be arbitrarily scaled, given the fact that all compute entities are deployed as clusters or as loosely attached entity in "passive" communication mode.
- *Communication scaling*: The communication bandwidth of each participating entity in space will be determined at launch time (assuming no in-space physical upgrades). As such, scaling is not possible. This is an important insight, because it deterministically leads to the conclusion that communication capabilities must be over-provisioned at launch-time to not only enable sufficient BW at initial phase P1, but also enable intra-SDC communication later as well as communication with more scout space-vehicles and consequently with higher BW with the ground-station.

- *Compute & memory scaling*: Once more, the amount of compute and memory capabilities of each participating entity in space will be determined at launch time (initially assuming no in-space physical upgrades). In analogy to the above, initial overprovisioning is highly recommended. It is worth noting that overprovisioning of memory is more important than overprovisioning of compute capabilities. The reason for this is the fact that idling services will consume memory in the clusters but virtually no compute capacity.
- *Deployment scaling*: With the concept of "each SDC is a cluster" for phase P1, transitioning to "all SDCs form a large cluster", the number of SDCs, and with that the number of scout space-vehicles, can scale arbitrarily. Worth noting in that context is that not all SDCs must have the exact same compute/memory/network chip generation: Over time, the compute payload of SDCs can follow the latest available chip generations. The reason for this is that clusters can seamlessly handle heterogeneous nodes.

Cost & energy optimization:

It is fair to assume that the overall cost for SDCs will be determined by the launch procedure. As a consequence and taking the insights from the previous sections into account, each SDC must be filled to the maximum weight with compute capabilities of the latest available generation (under the fair assumption that this will be most power efficient), and with memory overprovisioning and sufficient communication BW for end-of-life in deployment phase P2 (see previous paragraphs).

Besides the use of the latest generation compute components, the overall system design ensures operation at minimum energy consumption. The fundamental reason for this is that only the SDCs operate in "active" mode, all other components operate in "passive" mode. This leads to the fact that "something happens" (and consumes energy) only when "something is available".

Somewhat embarrassing that not more can be said here, but it's always best when the system design per-se automatically ensures operation at the lowest possible energy consumption.

10. SYSTEM SIMULATION (WP9)

The system simulator of this work is built as an MS-Excel spreadsheet that allows to enter specific design considerations and decisions and compare resulting figures of merit (FOMs). The resulting comparisons shall help to assess architectural and technological decisions and create and improve early blueprints for large, open space compute infrastructure.

The tool will not be sufficiently sophisticated to compute in detail performance metrices but is modular enough to allow the community to add the required level of details to be able to do so in the future. It is meant as a base for further development and will “live” from experts that will inject know how in the form of up-to-date data, equations and relations.

10.1. SIMULATOR ARCHITECTURE

A high-level schematic on how the simulator concept is built, is depicted in below figure. The input of the model consists of a “static” component, i.e. design decisions and system/workload requirements identified and defined by the user, and a “dynamic” component, which is derived from technology roadmaps, by sweeping over “time”. For every point in time (2023 – 2050), the model-output is stored as time-series of FOMs (which are also plotted). Additionally, for 1 point in time (to be selected by the user) a numeric text-output is provided for reference (box in green).

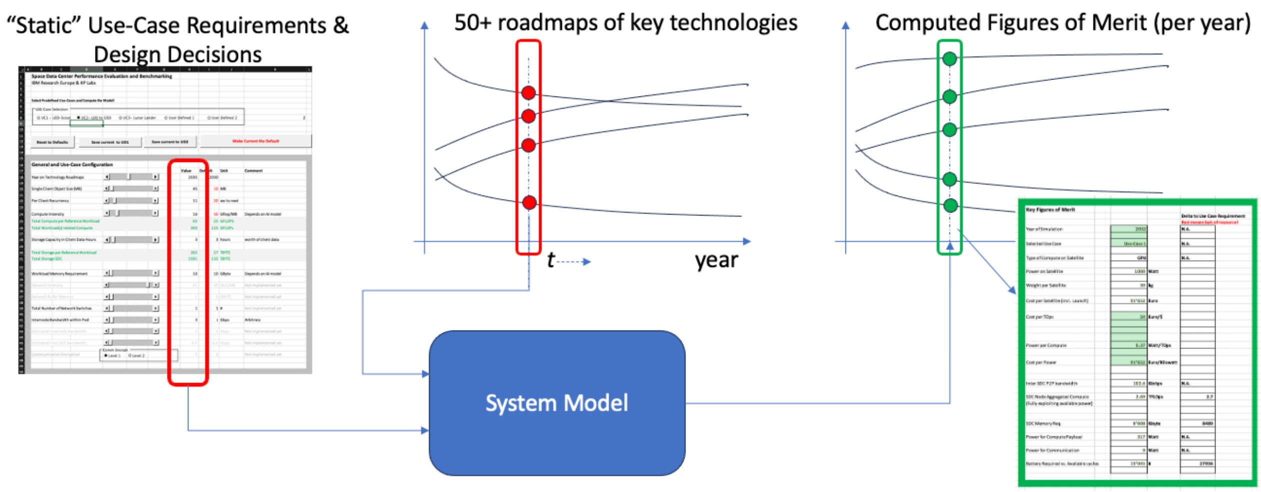


Figure 39 Simulator High-Level Architecture

A more detailed concept with the flow of information is provided in the following figure. Besides the individual models as described in the following sections, the “Name-Manager”, which is specific to MS Excel Spreadsheets, serves as an access point for all “global” variables from the spreadsheet, which are also accessed in the underlying VBA scripts, which are mainly responsible for setting up the data in the correct order for the models to consume. “temp_data” refers to aforementioned figures of merit (intermediate) data store.

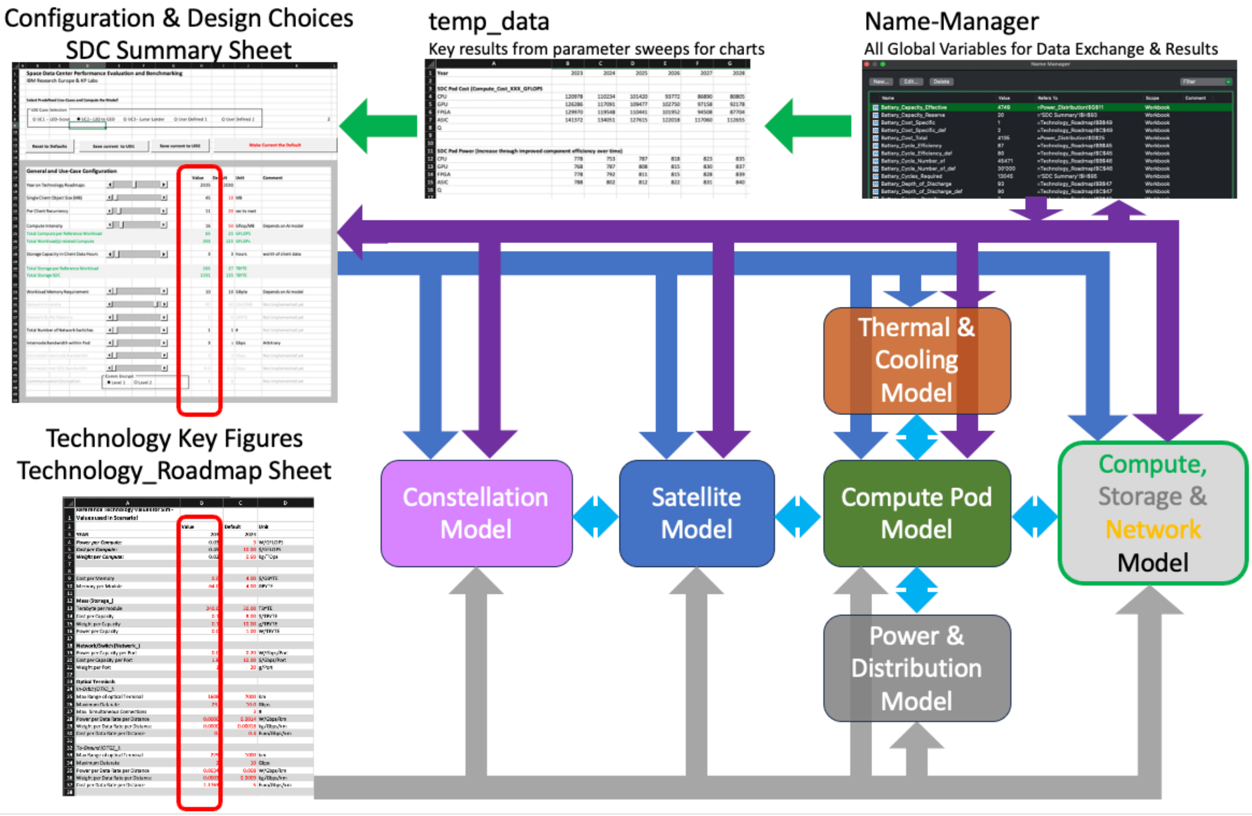


Figure 40 Simulator More Detailed Architecture with Dataflows

10.2. SIMULATION MODEL

As is seen from above figure, the actual model is hierarchically partitioned, from the top, i.e. the constellation, over individual satellites, with power, thermal and payload subcomponents, and finally the latter is implemented with separate compute, storage, and network “models”. Every (sub) model is implemented on a separate sheet of the main MS Excel workbook and data is exchanged solely through named ranges (variables), which are accessible globally. For practical and user-aspects of the simulator, the reader is referred to the “SDC Simulator User Manual.docx”.

It is important to note, that for dimensioning of the entire satellite, the power coming from the solar panels has been selected (***Power_Total_PV_Capacity***), thus once different overhead factors have been deduced, the available power determines the size of the compute payload and then propagates all the way up to the overall satellite size.

Throughout the next sections we will repeatedly refer to different sections of the user interface (UI), which is reproduced from the user manual in below figure.

Figure 41 User-Interface Overview with different sections.

Section 1:	Use Case selection and save-guarding of user made changes
Section 2:	Interactive section, where system requirements can be fine-adjusted or controlled
Section 3:	Time-series of different computed figures of merits
Section 4:	Interactive plot area for plotting mutual dependencies (over time)
Section 5:	Numeric Output window of key figures of merit for a given time-stamp
Section 6:	Static constants (for reference only)
Section 7:	Static chart on cost of compute at the edge versus raw-data download from space.

10.2.1. FROM USE-CASES AND CONSTELLATIONS

Different requirements from use-cases are stored on the ***Use-Case Definitions*** sheet of the tool. From there, they are automatically copied into the "Value" column of the UI on the ***SDC Summary*** sheet, which are the values used for evaluating the overall system model. On the "Value" column, individual quantities can be manually fine-adjusted if necessary and copied back to user-defined use-cases. Alternatively, changes can be made directly on the ***Use-Case Definitions*** sheet if the default use-cases are to be changed.

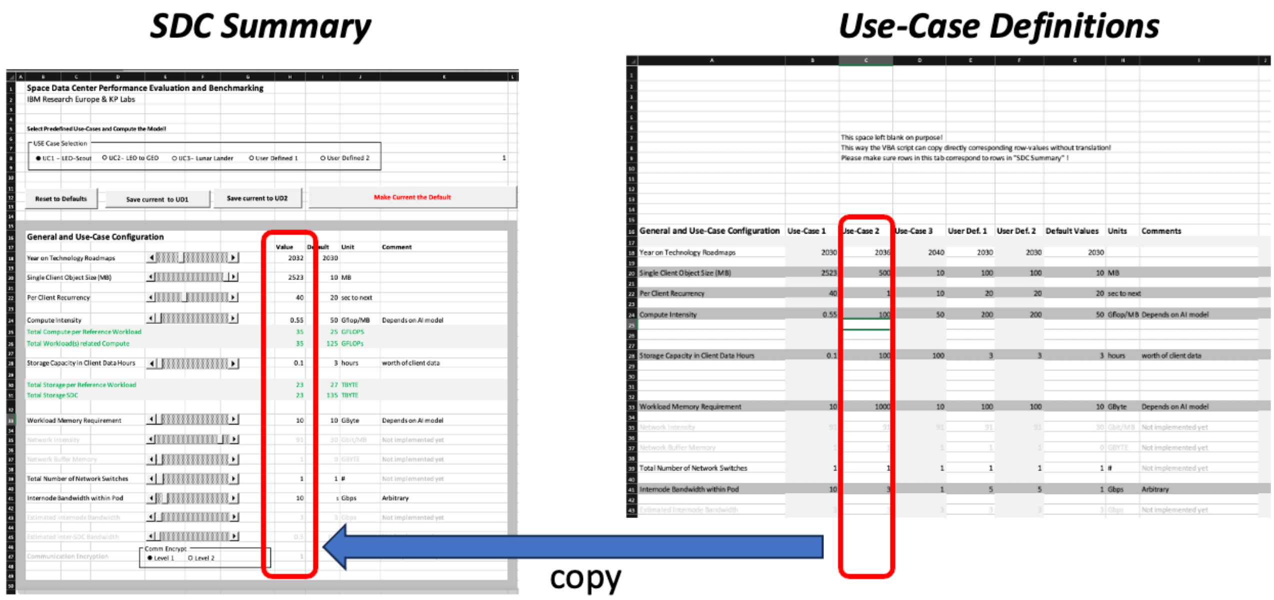


Figure 42 Use Case definitions are copied to the UI for model evaluation, triggered by selecting the radio-buttons in Section 1 of the UI

The constellation is of secondary priority in the current implementation of the model. On the ***SDC Constellation*** sheet, mostly satellite results are aggregated for reference. The following figure also depicts the general mechanisms used throughout the entire spreadsheet. To make computed values accessible to other sections of the spreadsheet, cells with formulas are assigned a cell-name (see top left). Besides being able to add meaningful names (instead of ***$G16*** the reference becomes ***Const_Total_Compute***), in case blocks of cells need to be shifted around, the reference to the named range remains intact, i.e. if a VBA code snipped references the named range, the code will remain intact. If otherwise the code would refer to a "native" cell coordinate like ***$G16***, that reference would remain and the code would brake!

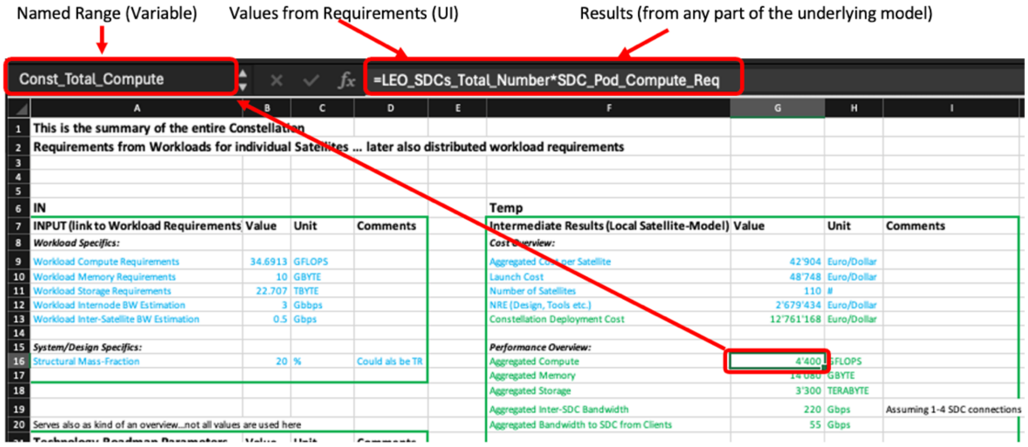


*Figure 43 Top section of the constellation module (**SDC_Constellation** sheet), depicting definition and use of named range variables.*

10.2.2. SYSTEM CONFIGURATION AND UI

Any simulation scenario is mainly configured in Section 2 of the UI. All values in the "value" column are assigned a named range (see previous figure), such that any changes on this sheet are directly reflected in the numerical results. This section is tightly linked to the use case definitions from the previous section.

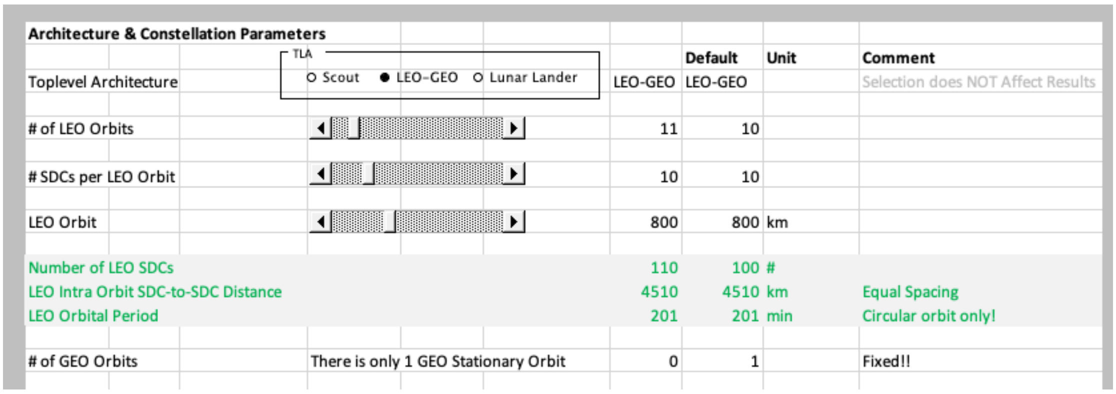


Figure 44 Example section of the UI configuration section.

Due to limitations of the (OSX) version of MS Excel and the size/complexity of this sheet, interactive slider movements may or may not work as intended. Alternatively, numerical values can be directly entered into respective cells. For every **change on the radio buttons**, respective values are copied into the active value column and **all results at the current timestamp are recomputed**.

10.2.3. TECHNOLOGY ROADMAPS

Technology development over time will be key to facilitate SDCs and make them economically viable in the future. Besides the discussion of these developments in the ***SDC Technology Roadmap Report V2***, the ***Technology_Roadmap*** sheet constitutes one, of not THE key-element of this system analysis tool. Over 50 different core technologies for SDCs have been quantified into roadmaps from 2023 up to 250.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Reference Technology Values for Sim - Values used in Scenario!**						**YEAR**	2023	2024	2025	2026	2027	2028	2029
2		**Value**	**Default**	**Unit**		**Performance Doubling Every XY Years**								
3	**YEAR**	2036	2023			red = exponential, yellow = linear								
4	***Power per Compute:***	0.006	0.1	W/GFLOPS	Assume Start in 2030	1.5	Quantum							
5	***Cost per Compute:***	0.006	0.10	$/GFLOPS		1.5	Quantum							
6	***Weight per Compute:***	0.031	0.50	kg/TOps		1.5	Quantum							
7														
8														
9	Cost per Memory	0.04	4.00	$/GBYTE		2.0		4.00	2.83	2.00	1.41	1.00	0.71	0.50
10	Memory per Module	80.63	4.00	GBYTE		3.0		4.00	5.04	6.35	8.00	10.08	12.70	16.00
11														
12	**Mass (Storage_)**													
13	Terabyte per module	285.41	30.00	TBYTE		4.0		30.00	35.68	42.43	50.45	60.00	71.35	84.85
14	Cost per Capacity	0.09	8.00	$/TBYTE		2.0		8.00	5.66	4.00	2.83	2.00	1.41	1.00
15	Weight per Capacity	0.11	10.00	g/TBYTE		2.0		10.00	7.07	5.00	3.54	2.50	1.77	1.25
16	Power per Capacity	0.05	1.00	W/TBYTE		3.0		1.00	0.79	0.63	0.50	0.40	0.31	0.25
17														
18	**Network/Switch (Network_)**													
19	Power per Capacity per Port	0.03	0.20	W/Gbps/Port		5.0		0.20	0.17	0.15	0.13	0.11	0.10	0.09
20	Cost per Capacity per Port	1.65	10.00	$/Gbps/Port		5.0		10.00	8.71	7.58	6.60	5.74	5.00	4.35
21	Weight per Port	13	20	g/Port		20.0		20	19	19	18	17	17	16
22														
23	**Optical Terminals**													
24	*In-Orbit (OTIO_):*													
25	Max Range of optical Terminal	17236	7000	km	SCOT80	10.0		7000	7502	8041	8618	9237	9899	10610
26	Maximum Datarate	24.6	10.0	Gbps		10.0		10.0	10.7	11.5	12.3	13.2	14.1	15.2
27	Max. Simultaneous Connections	7	3	#	Allow fraction of connecti	10.0		3	3	3	4	4	4	5

Figure 45 Technology Roadmap Overview

Most roadmaps are assumed to be exponential, with the time within which performance doubles defined in column "F". Some are also assumed to be liner, visible through the yellow font in column "F", instead of red. The starting value for the roadmap is defined in the default column "C". As there is NO 1 technology roadmap for every single of the used quantities, **default values and growth-rate have been selected based on available knowledge**, reading articles and discussions with experts. Some may or may not be justified and/or in full agreement with the community **and may be changed to reflect the users domain expertise or conviction**.

In the "*Update_Year(year)*" subroutine in the VBA code, the columns H and following are copied to column "B", as a function of the year, and thus made available to the model. All values in column "B" are referred to with named ranges, thus globally available to the model.

10.2.4. SDC SATELLITE

The satellite “Model” is mostly an aggregation sheet of the underlying models (see Figure 40).

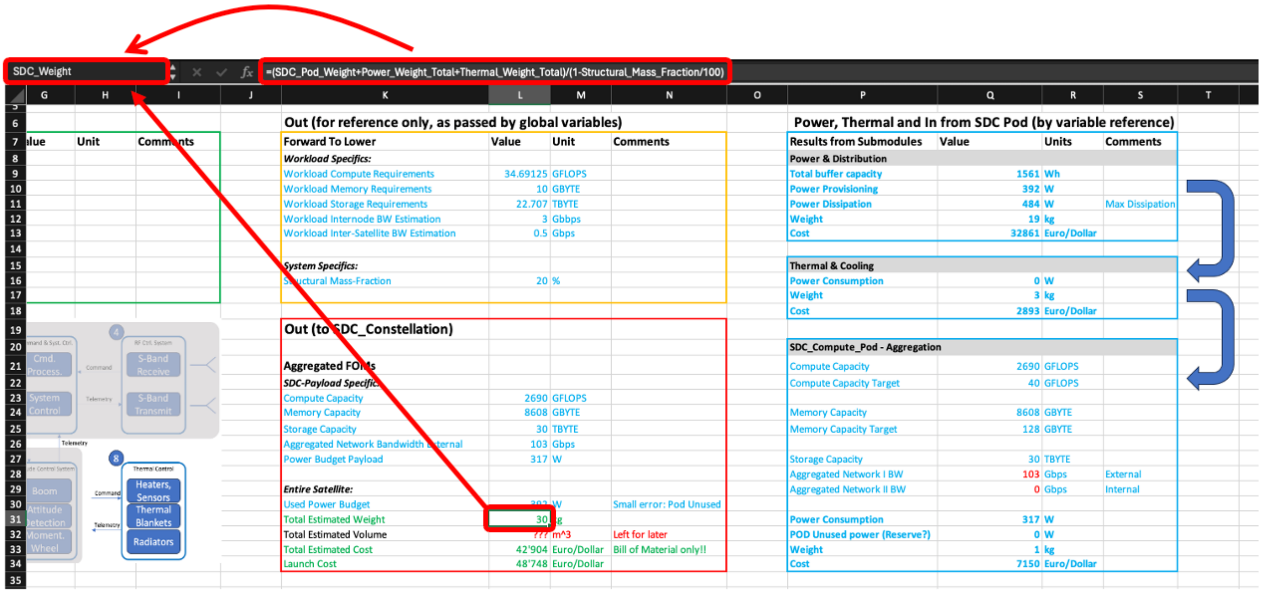


Figure 46 SDC Satellite “Model” Overview”

Above figure again depicts the use of named ranges to compute and refer to values and results from other sections of the model.

10.2.5. SDC POD

The pod of an SDC is considered the payload module with all compute-, networking- and storage capabilities. In this sub-model, satellite requirements are broken down into node requirements (we assume here a single compute pod with several/many nodes. For future scenarios with larger satellites, this approach can be extended towards several pods per satellite too.

As indicated with arrows, from workload/satellite requirements, the node-requirements are derived and provided to the “***Compute_Storage_Network***” model, where compute capacity, network and communication performance is computed according to available (power) resources.

The results from the latter are then aggregated to satellite level and returned to “***SDC_Satellite***”.

This module is important as on one side it derives important quantities from the workload requirements and from the configuration input, and on the other hand it aggregates values from the entire payload to be represented on the satellite level.

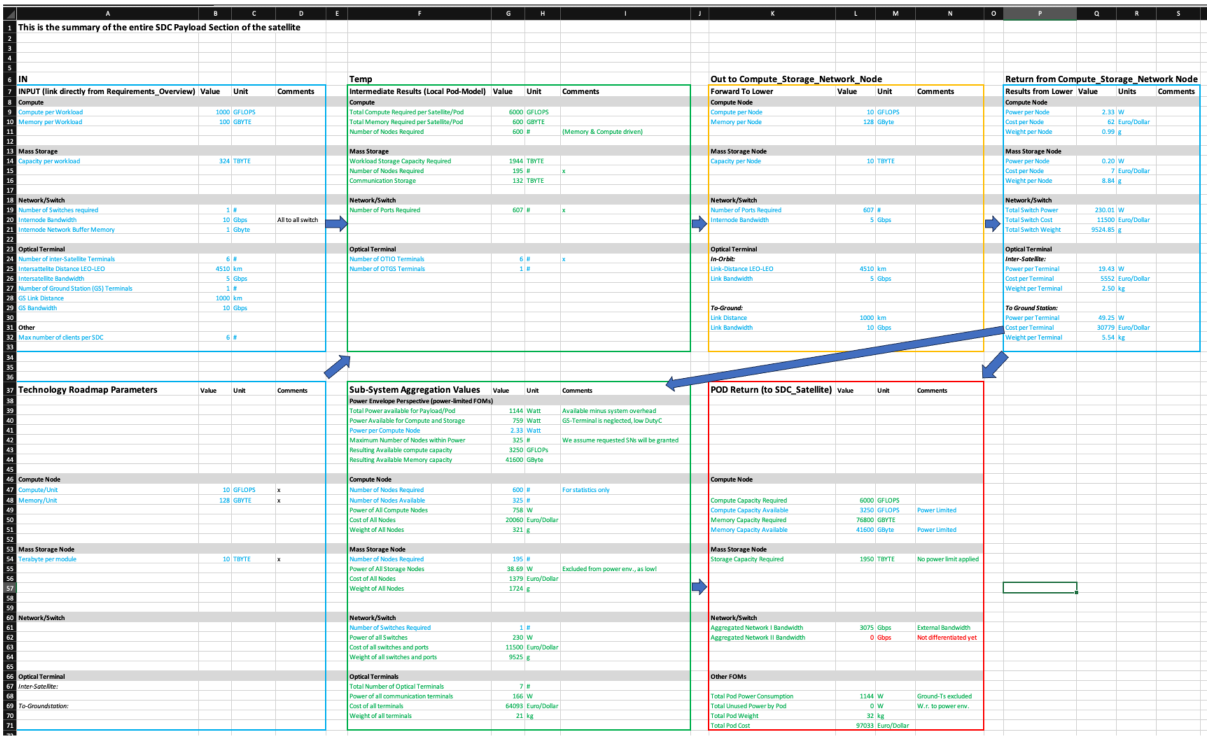


Figure 47 SDC Pod sheet overview

10.2.6. POWER AND POWER DISTRIBUTION

This model is based on assumptions of conversion efficiency, battery technology (e.g. depth of discharge [DoD]) and orbital parameters (duration of night -> battery capacity required). The sub-model key output is available power to the compute pod and cost and weight of the battery, solar panel, and power-conversion systems.

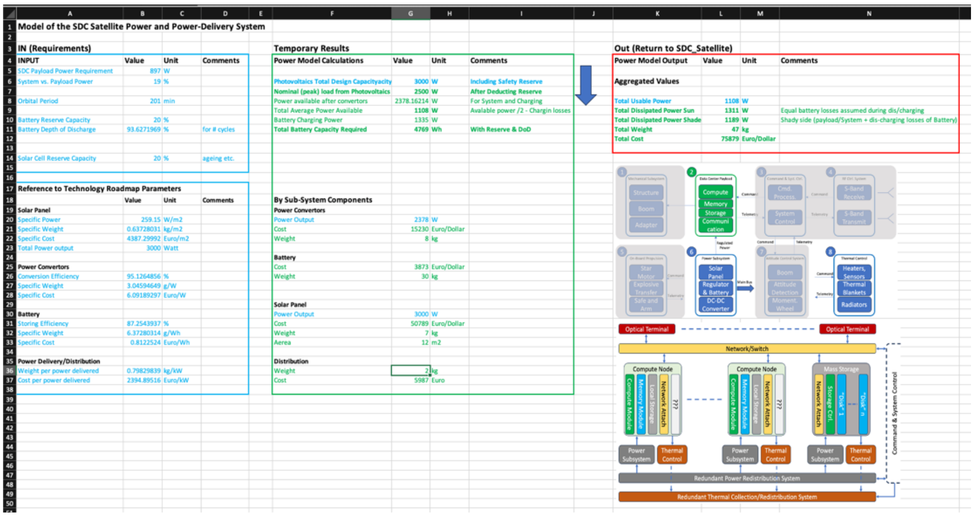


Figure 48 **Power Distribution** sheet

10.2.7. THERMAL AND COOLING

The main contribution of this sub-model is the derivation of the cooling requirements, mostly weight, secondary also cost, to be able to operate at the given power-envelope. Only less than half of what the solar panels provide at peak, will be available for satellite system and payload operation, as the other half is needed to re-charge the batteries, such that the satellite is always operational, irrespective of day- or night-time. While this may be questioned, as most observation-data is acquired during daytime, our assumption constitutes the most generic scenario, which with emerging hyper-spectral and SAR sensors will become more relevant. Also, any node in the constellation may be assigned tasks, such as compute and communication, irrespective of solar exposure, i.e. where they are w.r. to their orbital cycle.

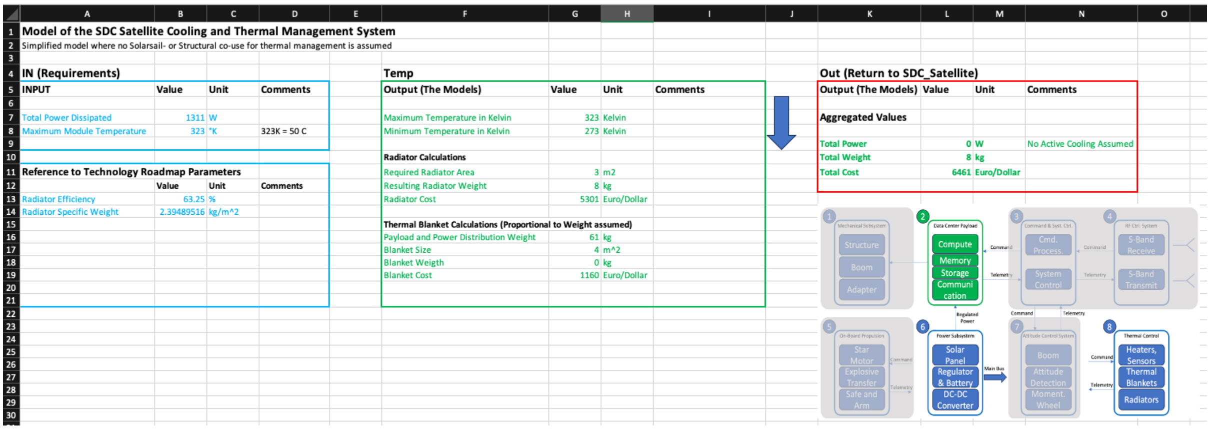


*Figure 49 **Thermal_and_Cooling** sheet overview*

10.2.8. COMPUTE, COMMUNICATION AND STORAGE

This sub-model is key for the entire model. Never-the-less, the actual models of the compute node, communication terminals etc. are very simple and consist mainly of multiplying the node-requirements with the “efficiency” (power specific compute, weight specific compute etc.) numbers from the current timestamp of the technology roadmap.

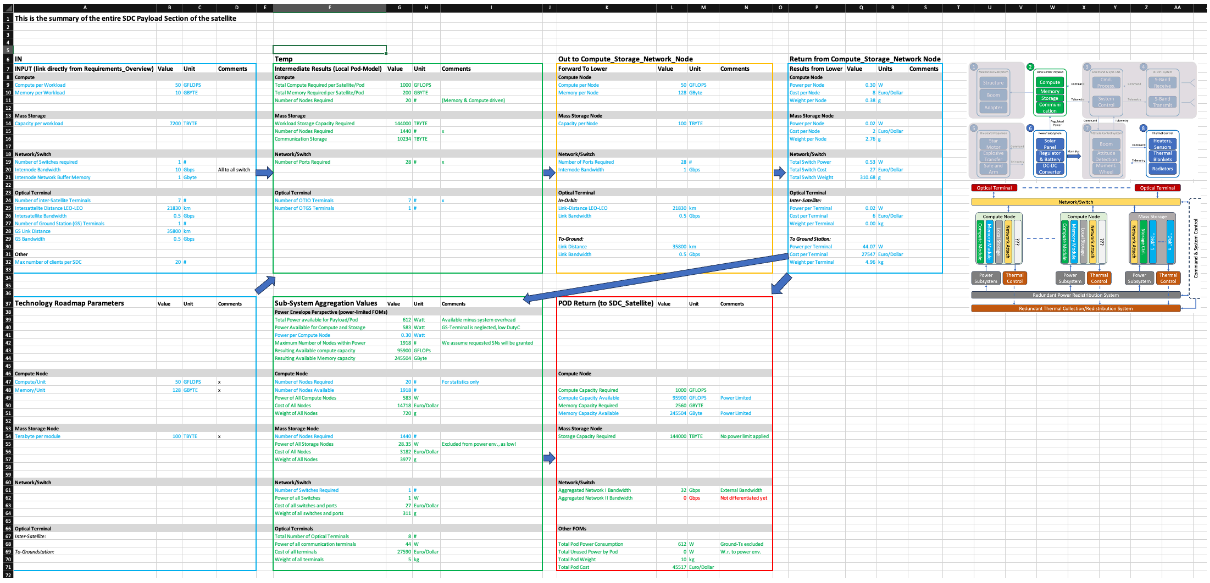


Figure 50 ***Compute_Storage_Network*** *sheet overview*

10.2.9. SOFTWARE STACK, OS, AND NETWORK ARCHITECTURE

While these three topics are foreseen for future integration into the analysis tool, they would require specific testing and evaluation for various scenarios, which is beyond the scope of this study. All the overhead is thus absorbed in bulk into the overall system-overhead and other reserves.

10.3. TESTING AND VALIDATION

To our knowledge, there is no comparable tool available (in the open), thus benchmarking of accuracy is not possible. Nevertheless, testing and validation was performed by means of "reasoning". The following steps were conducted:

A. All Technology Roadmaps have been plotted individually to verify continuity and plausibility.

B. Type-errors have been introduced, e.g. specific numerical value on UI have been replaced with a string (e.g. *Power_Total_PV_Capacity* from int(3000), to str("Error")), to track hierarchical error propagation and dependencies between sheets.

C. Extensive debugging on the VBA side, using "Debug.Print" statements to track algorithmic progression in the MS Excel immediate window.

D. Plausibility checking (order of magnitudes) against existing systems (e.g. Starlink)

10.4. SAMPLE ANALYSIS WORKFLOW (TUTORIAL)

For a step -by-step tutorial, please see ***SDC Simulator User Manual.docx***., section “Example System Analysis”.

10.5. USE CASE SPECIFIC SIMULATION RESULTS

In this section we present some “key” findings from simulating the different, pre-defined use-cases and discuss specific assumptions to be able to map the use-cases to the simulator in a unified way.

10.5.1. USE CASE 1 – MOTHERSHIP AND SCOUT SATELLITE

Description: In this use-case, a smaller observation satellite orbits synchronously to a LEO SDC, with constant connectivity to the SDC. The SDC receives wide field of view observation data and tries to detect wildfires. If the SDC identifies potential regions of interest (ROIs), it aligns its own, narrow field of view, but high-resolution observation instrument to the ROIs to acquire high resolution data and provide actionable insights to clients through a ground station.

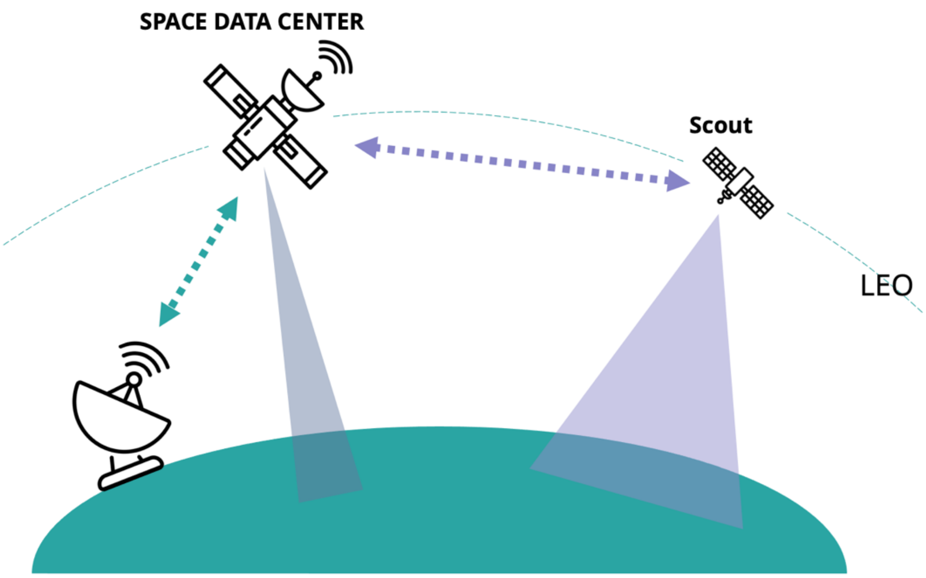
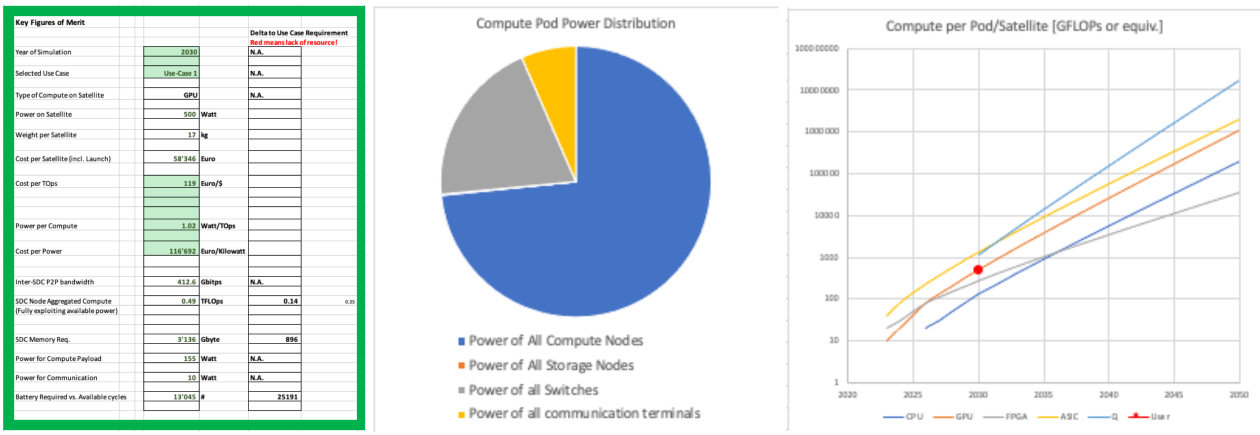


Figure 51 Use Case 1: Wildfire – Mothership-Scout Configuration

Assumptions:

1. Continuous connection between scout and SDC satellite (by means of optical terminals).
2. Although only a single SDC is considered, we assume a constellation of SDCs which allows data to be continuously downloaded to ground stations. The constellation aspect, however, is mostly neglected.
3. Equivalent framerate of 460 x 460 x 3 pixels per second, for both, large- and narrow field of view instruments (see Section 7.2).
4. Computational cost is 550 GFLOP/MByte of observation data (see Section 7.2).
5. Regions of interest in large field of view images is very rarely observed, such that the additional computational load on the SDC can be neglected.
6. For practical reasons we assume a time-horizon of 2030 and GPUs as the compute engines.

Results & Interpretation:



Key Figures of Merit			Delta to Use Case Requirement Red means lack of resource!	
Year of Simulation	2030		N.A.	
Selected Use Case	Use-Case 1		N.A.	
Type of Compute on Satellite	GPU		N.A.	
Power on Satellite	500	Watt		
Weight per Satellite	17	kg		
Cost per Satellite (incl. Launch)	58'346	Euro		
Cost per TOps	119	Euro/$		
Power per Compute	1.02	Watt/TOps		
Cost per Power	116'692	Euro/Kilowatt		
Inter-SDC P2P bandwidth	412.6	Gbitps	N.A.	
SDC Node Aggregated Compute (Fully exploiting available power)	0.49	TFLOps	0.14	0.35
SDC Memory Req.	3'136	Gbyte	896	
Power for Compute Payload	155	Watt	N.A.	
Power for Communication	10	Watt	N.A.	
Battery Required vs. Available cycles	13'045	#	25191	

Figure 52 Use Case 1: Scout-Mothership – Results

As can be seen from the workload requirements, intuitively, the workload for dedicated compute hardware is rather moderate. With a system power-envelope of only 500 Watt, a tiny satellite weight of 17 kg results, which easily accommodates the required 350 GFLOPs, actually 490 GFLOps, thus ~ 25 % reserve. With a single scout as a client, this use-case defeats an important purpose of SDCs, to share resources within a constellation and is only justified by the time-lag between the two instruments, which allows sufficient time to re-align the high-resolution instrument to ROIs.

Clearly an oversimplification in this scenario, is the relatively low power-consumption of the optical communication, which stems from the low required data-rate of “actionable” insights. For the linear model of the power- and cost-efficiency of optical communication terminals to be more accurate, data-rates should likely be closer to the value of ~ 5 Gbps, which was used to derive the linearization, instead of assumed 0.5 Gbps. Never-the-less, the results seem to be meaningful and the power-break-down into compute nodes, communication, and switches, makes just about sense.

Another interesting observation is cost-development over years for the assumed GPU compute technology. While compute per pod will exponentially increase (given a fixed power envelope), the cost per pod will have a minimum around 2035. This can be explained with two differing exponential growth-rates assumed: 1.9 for performance and 2.7 for cost. Whether these are justified or not is at the user’s discretion.

Finally, it can be seen, that significant power goes into network switches. This is because only 10GFLOPs are assumed to be available per node, which yields a total of ~ 35 nodes, thus 35 ports for the single main switch we assumed. As power is assumed to scale linearly with the number of ports, this large power fraction is observed. Higher compute per node, would strongly reduce this contribution.

10.5.2. USE CASE 2 – MULTIPLE LEO CLIENTS TO GEO SDC

Description: In this use-case, a scenario is assumed, where LEO observation satellites upload their observation data directly to an SDC on a GEO orbit. In Figure 53 a slightly more generic scenario is depicted, where an SDC constellation on LEO can also receive client data and uses the GEO SDC only in situations where no viable downlink paths to ground-stations exist.

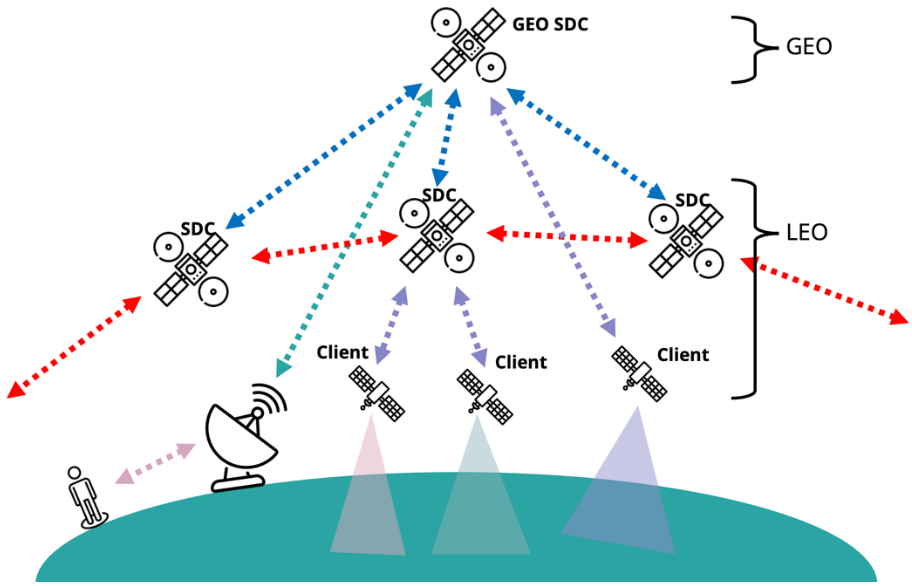


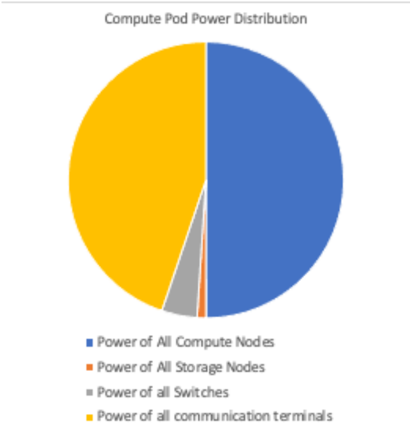
Figure 53 Use Case 2: Generic LEO – GEO EO Observation Constellation

Assumptions:

1. Continuous connection between GEO SDC and the ground-station.
2. For the lack of a better workload, we assume use-case 1's wildfire application, just extending it to multiple LEO observers.
3. The equivalent framerate is then 460 x 460 x 3 pixels per second (see Section 7.2).
4. Computational cost is 550 GFLOP/MByte of observation data (see Section 7.3).
5. As there are no SDCs in the base use-case, # LEO Orbits and #SDCs per LEO Orbit are "0"
6. We assume 5 concurrent clients per GEO SDC and 3 GEO SDCs to cover along the entire equator.
7. To increase flexibility and exploit the GEO position, a relatively large storage capability of 100 hours of client data is assumed.
8. As no LEO-to-LEO communication is accounted for, client to SDC distance is replaced with the GEO orbit altitude (see cell $G22 on the ***Compute_Storage_Network*** sheet)

Results & Interpretation:

Key Figures of Merit			Delta to Use Case Requirement Red means lack of resource!	
Year of Simulation	2034		N.A.	
Selected Use Case	Use Case 2		N.A.	
Type of Compute on Satellite	GPU		N.A.	
Power on Satellite	1000	Watt		
Weight per Satellite	56	kg		
Cost per Satellite (incl. Launch)	1'301'413	Euro		
Cost per TOps	329	Euro/$		
Power per Compute	0.25	Watt/TOps		
Cost per Power	1'301'413	Euro/Kilowatt		
Inter-SDC P2P bandwidth	413.5	Gbitps	N.A.	
SDC Node Aggregated Compute (Fully exploiting available power)	3.95	TFLOps	2.20	1.75
SDC Memory Req.	10'112	Gbyte	5632	
Power for Compute Payload	295	Watt	N.A.	
Power for Communication	192	Watt	N.A.	
Battery Required vs. Available cycles	13'045	#	30877	



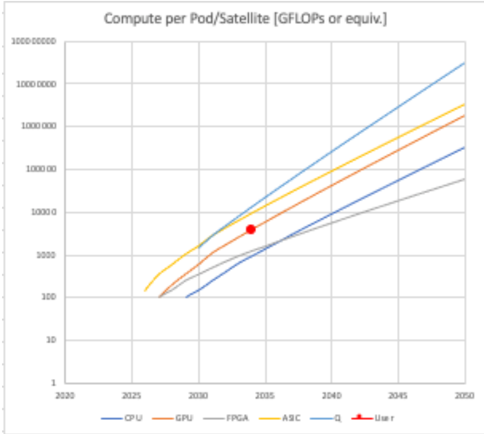


Figure 54 Use Case 2: LEO Observation, GEO SDC - Results

Again, the workload requirements for dedicated compute hardware are rather low. With a system power-envelope of 1000 Watt, a satellite weight of 56 kg results, which easily accommodates ~4 Teraflops in 2034, twice as much as required.

Compared to use-case 1, we notice a significantly higher power requirement from communication, which of course is explained by the ~ 36x longer communication distance from the SDC to ground-stations. As overall data-rate is still moderate, even 100 hours of incoming data-storage, are negligible on the power-draw.

From the compute per Pod curve, we notice missing data before the year 2025. This is because communication and network/switch power consume all available 1000 Watt, thus no compute can be supplied.

10.5.3. USE CASE 3 – LUNAR LANDER

Description: In this use-case, the LEO clients are substituted by moon rovers and the LEO or GEO SDC replaced with an SDC within the lunar lander. After extracting information from the rover data, insights and compressed data is sent to earth. For visibility reasons, a relay satellite at the L2 Lagrange point transmits data to earth.

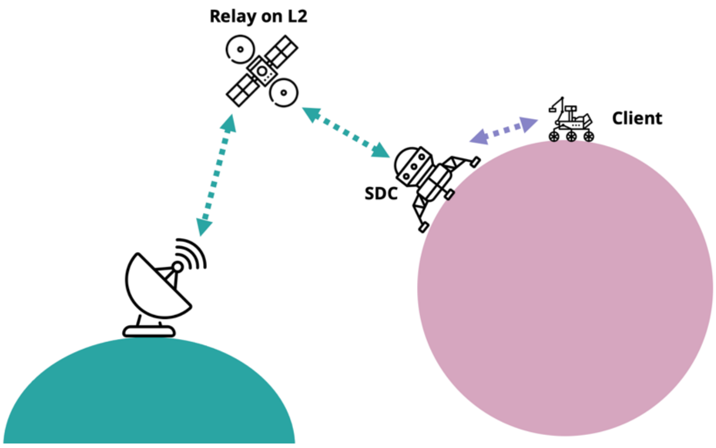


Figure 55 Use Case 3: Lunar Lander

Assumptions:

1. To map the scenario onto the capabilities of the spreadsheet simulator, the "Lunar-Lander radio-switch" replaces the LEO intra/inter orbit switch with 100km fixed. While the key-performance metrics on the communication terminals are for optical links, we assume RF-links between the rovers and the SDC, but neglect differences in metrics between RF and optical. This could be done in the equation defining the ***OTIO_SDC_Power*** named range.
2. To implement the optical link from the moon to the relay on L2, the ground-connection of the SDC is replaced with the distance between the moon and the L2 point.
3. Any resource beyond the SDC on the lander, e.g. the relay satellite on L2 is NOT modelled in this environment. It is assumed such infrastructure could be shared between many similar missions.
4. As with the previous use-cases, there is a lack of clearly defined workloads. To overcome this, we use the previously discussed workloads again as a reference but reduce the expected observation data-object (i.e. image) to the size of 1000 x 1000 x 3 pixels (i.e. 3 MByte, see also Section 7.4) and further assume a 1-minute repetition rate.
5. We assume a "large" fleet of 20 concurrently operated lander in this scenario.
6. Computational cost is 550 GFLOP/MByte of observation data (see Section 7.2).
7. With no means of servicing the infrastructure, we foresee a large (100 client hours) storage.

8. For workload flexibility, also for this use-case, GPUs are assumed to be the compute workhorse.

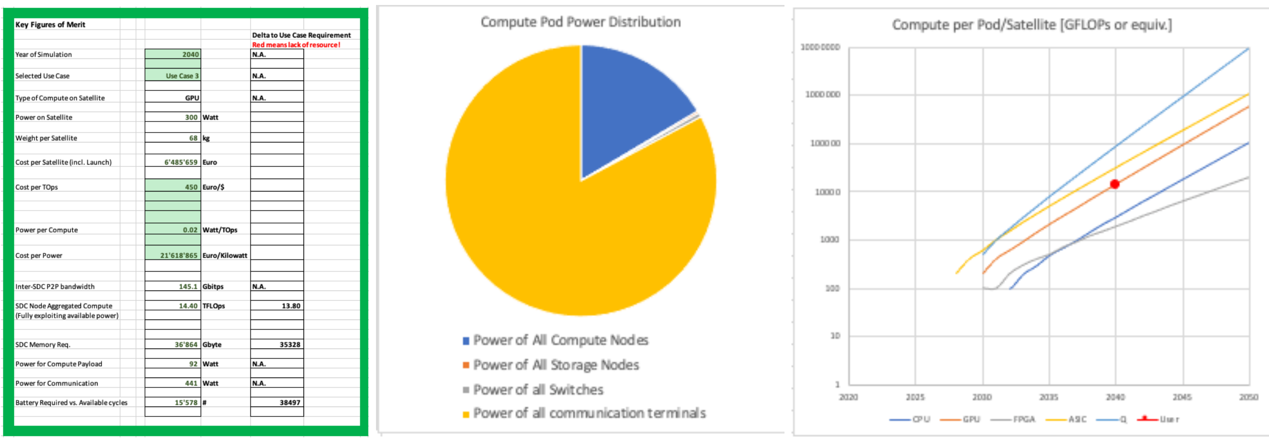
Results & Interpretation:

To get infrastructure to the moon is expected to be much more costly, also in the use-case reference year 2040, as it will have to be landed safely on the lunar surface. It is thus good to see that one side way more compute will be available to also accommodate future workloads like e.g. federated or “remote” learning of algorithms, but also that the expected power-envelope of only 300 Watt yields a relatively compact satellite of only 68 kg (of course this does not include specific moon-landing infrastructure like fuel and thrusters).

From Figure 56 we see that communication, given the large distances, contributes with the bulk of the power-consumption.

Even though the relative storage capacity of 100 hours is large, the small data-payloads and recurrency rate yields relatively small overall storage, which is also reflected in it’s manageable if not negligible power-consumption.

As in the previous use-case, data before ~ 2030 is not available, as with this power envelope, with the large communication overhead, there is not enough power for compute, despite advancements in compute and other technologies.



Key Figures of Merit			Delta to Use Case Requirement Red means lack of resource!
Year of Simulation	2040		N.A.
Selected Use Case	Use Case 3		N.A.
Type of Compute on Satellite	GPU		N.A.
Power on Satellite	300	Watt	
Weight per Satellite	68	kg	
Cost per Satellite (incl. Launch)	6'485'659	Euro	
Cost per TOps	450	Euro/$	
Power per Compute	0.02	Watt/TOps	
Cost per Power	21'618'865	Euro/Kilowatt	
Inter-SDC P2P bandwidth	145.1	Gbitps	N.A.
SDC Node Aggregated Compute (Fully exploiting available power)	14.40	TFLOps	13.80
SDC Memory Req.	36'864	Gbyte	35328
Power for Compute Payload	92	Watt	N.A.
Power for Communication	441	Watt	N.A.
Battery Required vs. Available cycles	15'578	#	38497

Figure 56 Use Case 3: Lunar Lander - Results

10.6. KNOWN LIMITATIONS AND OUTLOOK

For user-level level limitations and issues, please refer to the ***SDC Simulator User Manual .docx***.

Clearly, the current implementation of this spreadsheet (v6) must be considered an initial version, with basic functionality. The vision is that this can serve as a base to further include more detailed models of:

a) Architectural decisions

b) Network architecture aspects

c) Security and encryption options, both within communication and computing

d) Operating system and software stack modeling

e) Moving from system analysis toward system optimization, by providing performance targets (other than the current power envelope) and few boundary conditions

Many of above points have already been accounted for in the analysis or the UI, but require further effort to implement in a generally usable way.

11. SUMMARY AND OUTLOOK

In this document we started by summarizing system-level requirements from the corresponding reference document and with those in mind, studied all aspects of space data center system architecture, from existing constellations, over edge-computing and data-center architectures, network considerations specific workload analysis, software stack considerations to satellite hardware implementation. By studying the corresponding satellite systems and constellations, we were able to derive a suitable simulation model structure to hierarchically derive key figures of merit for different use-cases, based on various technology roadmaps and additional user-requirements.

By combining traditional Microsoft Excel Spreadsheet best practice, with additional visual basic macros, we were able to build a powerful tool to early assess various predefined scenarios but also allow the user to define and simulate his/her own scenarios and easily share them with the community through a broadly available tool (Microsoft Excel).

Results based on many technological assumptions appear to be reasonable, but foremost depict interesting trends and dependencies, which is the main goal of a tool with a “forecasting” horizon 2050.

One of the main goals of this work and of the simulation tool, was to understand, if and with what boundary conditions space data centers and AI edge-computing in space will provide viable solutions for the future of the New Space Economy. In that view, we also wanted to understand if extracting information at the edge can be financially more viable than sending raw-data to earth, even with the emerging/potential availability of high-bandwidth space communication, as may be expected, and has already been demonstrated, by companies like Starlink or OneWeb. Based on the spreadsheet results, a “static” graph was derived and is displayed in Section 7 of the UI. We tried to use rather conservative values for the performance of edge-computing and aggressive values for communication, to get a conservative estimation of trends. The figure is reproduced in Figure 57 and nicely shows that there is a good chance, that eventually edge-AI will yield lower cost overall solutions, even with available high-bandwidth data-communication services.

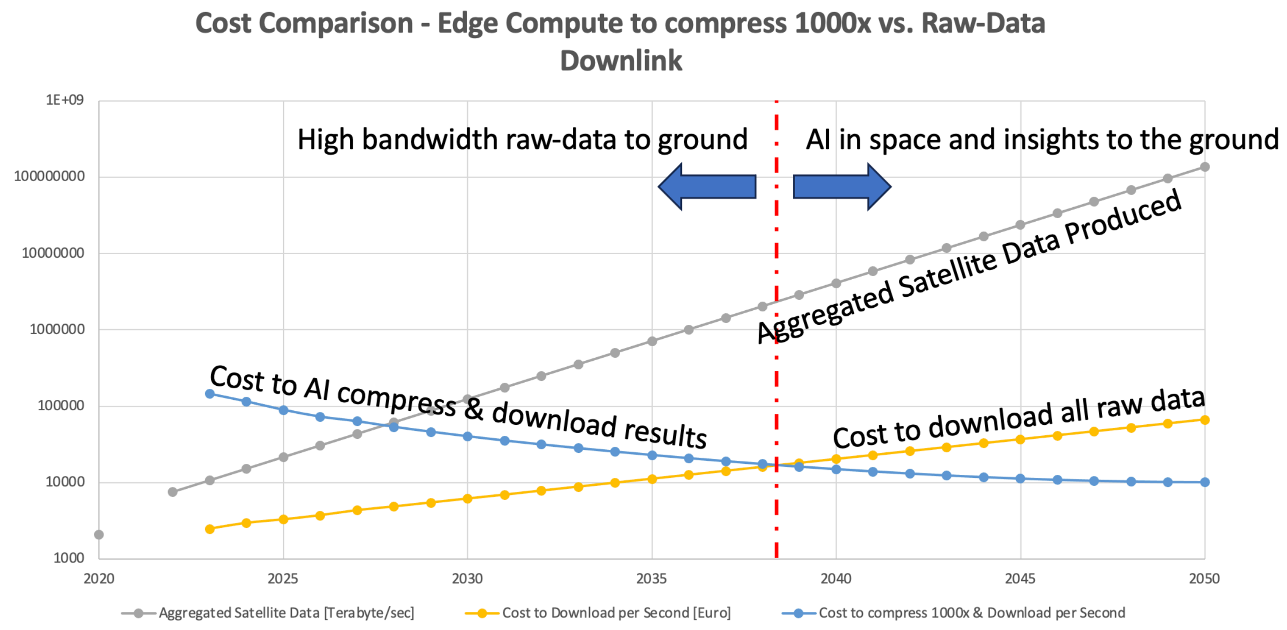


Figure 57 Cost Comparison between AI edge-computing and high-bandwidth raw-data communication.

Important assumptions made to create above figure are:

1. Lifetime of the "compute" hardware at the edge is 6 years (= value depreciation included)
2. Most recent trend in satellite deployment was for communication (Starlink, OneWeb etc.), thus we do NOT expect the aggregated data produced in space to scale identically as the numbers of satellites deployed.
3. To account for communication satellites, we thus expect the number of data-producing satellites to double every 8 years
4. We assume similar U-Net semantic segmentation AI architecture to achieve a 1000x compression of raw-data (see rows 75 and following on the ***Technology_Roadmap*** sheet.

As by contract, this, and all accompanying documents, as well as the simulation tool itself will be made publicly available, we hope that the community will see value in our work and adopt, adapt and re-use as much as possible of it. We will be happy to support such efforts within the realm of what's possible.

12. REFERENCES

[1] https://en.wikipedia.org/wiki/Satellite_constellation

[2] https://en.wikipedia.org/wiki/Ground_station

[3] Hamid Hemmati, Deep Space Optical Communications, JPL, CalTech, https://descanso.jpl.nasa.gov/monograph/series7/Descanso_7_Full_Version_rev.pdf

[4] Optical Communications Terminal (OCT) Standard Version 3.0, https://www.sda.mil/wp-content/uploads/2022/04/SDA-OCT-Standard-v3.0.pdf

[5] C. Cuerno-Rejado et al., Rapid Method for Spacecraft Sizing, https://doi.org/10.1243/PIME_PROC_1995_209_286_02

[6] The NIST Cybersecurity Framework 2.0, Initial Public Draft, https://doi.org/10.6028/NIST.CSWP.29.ipd [Accessed 25.10.2023], August 2023

[7] M. Scholl, T. Suloway, Introduction to Cybersecurity for Commercial Satellite Operations, NIST IR 8270, https://doi.org/10.6028/NIST.IR.8270 [Accessed 25.10.2023], July 2023

[8] NSA Cybersecurity Advisory, CNSA 2.0, https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF [Accessed 25.10.2023], September 2022

[9] A Novel 3D DRAM Memory Cube Architecture for Space Applications

[10] https://www.microchip.com/en-us/product/SST26LF064RT

[11] Tbit/s line-rate satellite feeder links enabled by coherent modulation and full-adaptive optics, https://www.nature.com/articles/s41377-023-01201-7

[12] S. Wang, X. Hou and X. Zhao, "Automatic Building Extraction From High-Resolution Aerial Imagery via Fully Convolutional Encoder-Decoder Network With Non-Local Block," in IEEE Access, vol. 8, pp. 7313-7322, 2020, doi: 10.1109/ACCESS.2020.2964043.

[13]