Phoenix Systems

developer of Phoenix-RTOS

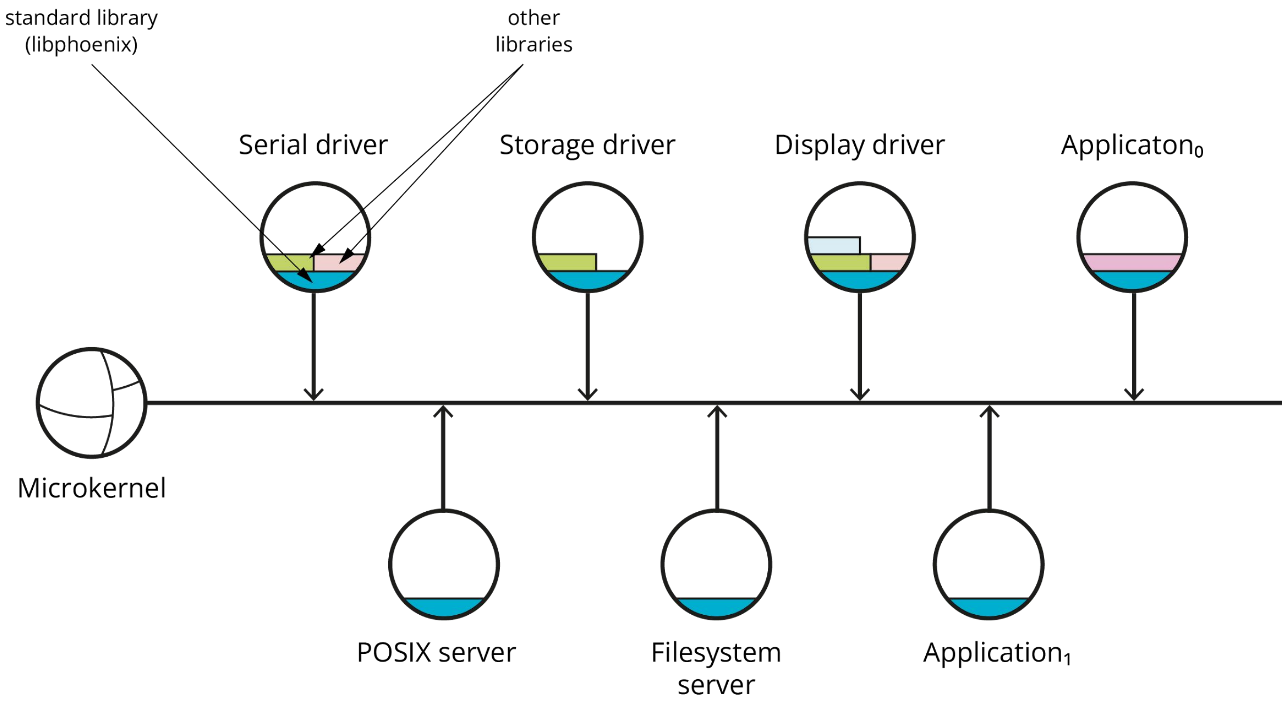
Edge-IoT computing in space exploration

Paweł Pisarczyk

Phoenix-RTOS

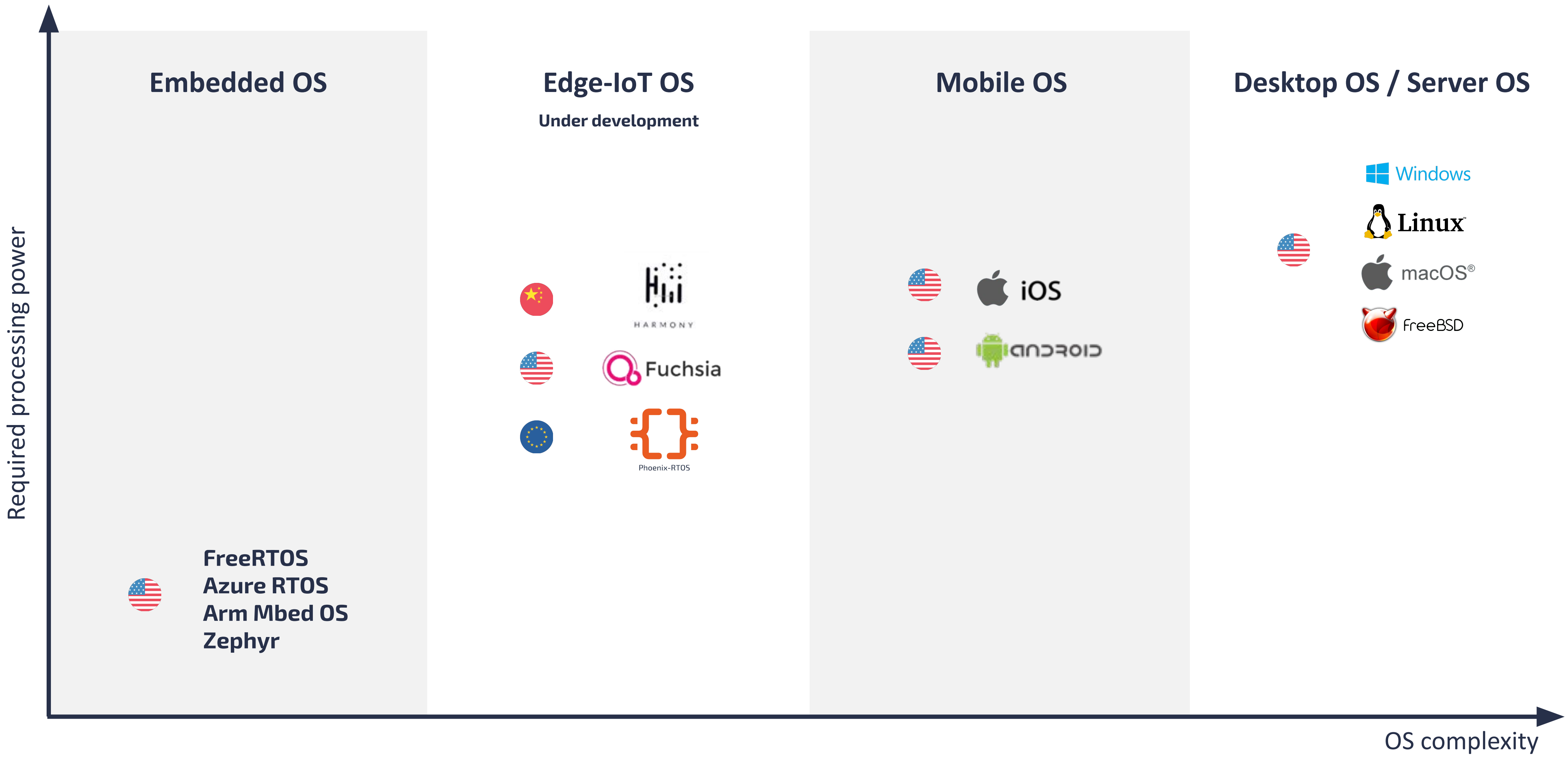
- Open-source operating system for Edge-IoT devices
- Scalable microkernel architecture
- Based on written from scratch microkernel
- Source code available under BSD license on GitHub: github.com/phoenix-rtos
- Prototype developed at Warsaw University of Technology (1999)
- Allows to run application on IoT devices transforming them into Edge-IoT devices
- Scalability from microcontrollers to advanced multiprocessor system (supported architectures: x86, ARM, RISC-V, LEON3FT)
- Support for software development safety assurance methodologies (e.g. DO-178C) required for mission critical application
- Frameworks simplifying development of Edge-IoT devices for many industry sectors

Phoenix-RTOS – microkernel architecture



- Operating system is implemented as the set of communicating components
- Microkernel provides basic resource management functions – memory management, processing management (processes and threads), inter-process communications
- Microkernel forms virtual bus between components
- Device drivers, filesystems, communication stacks are implemented as user-level servers
- Messages are exchanged synchronously using optimization preventing data copying
- Efficient modularity and scalability (no "spaghetti" dependencies)

Operating system market landscape



Phoenix-RTOS implementations – Smart Grid

Data Concentrator Unit with balancing meter - **60K devices** implemented in Energa-Operator (DSO)

Apator iSmart1 smart gas meter - **16K devices** implemented in PSG (Poland)

besmart.energy smart energy meter – prototype of smart meter with energy management, **license: open-hardware**

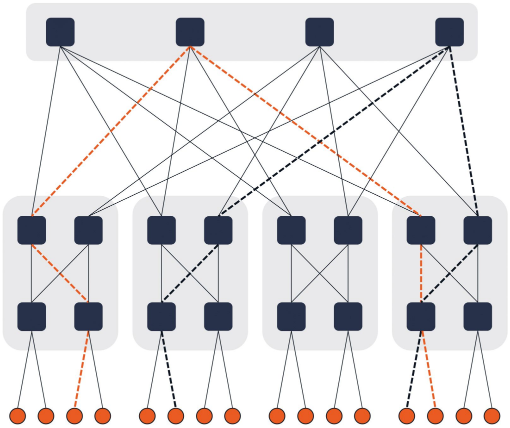
Apator iSmart2 smart gas meter - **1M devices** for Fluvius System Operator CV (Belgium)

Apator NILEE smart energy meter – smart energy meter with energy management functions

Space – current developments

Phoenix-RTOS has been ported for RAD-hard SoCs based on LEON3FT and NOEL (GR716, GR712RC, GR765)

- for software-defined CubeSats for deep-space missions,
- for scientific instruments development for deep-space missions,
- for ECSS qualifications.

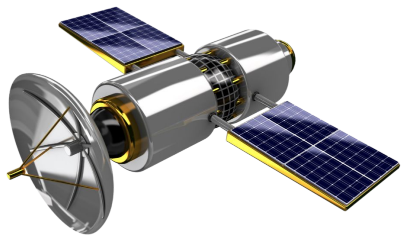


Product definition phase of Phoenix-RTOS SPACE HPC – distributed operating system version for space HPC/MPP applications

Phoenix-RTOS SPACE HPC project definition

Project goal

- **Development of Phoenix-RTOS SPACE HPC - distributed operating system for HPSC**
- Support for European high-performance RAD-hard SoCs
- Development of reference designs and example applications in collaboration with platform vendors (Frontgrade Gaisler, nanoXplore), space research institutions (CBK), space agencies (POLSA, ESA) and final users
- Support for emerging high-performance communication interfaces e.g. SpaceFibre, RapidIO, Ethernet TSN, SpaceVPX
- Preparation for ECSS qualification



Project motivation

- **Lack of operating system for space applications enabling development of advanced, multi-component software systems**
- Bunch of application development frameworks developed by space agencies (e.g. NASA cFS, JPL F’) enriching the functionality of commonly used RTOS-es (e.g. RTEMS, VxWorks) and preserving portability
- Inability to use commonly used RTOS-es as the basis for space MPP system
- Inability to use GNU/Linux in deep space missions because of its monolithic and unscalable architecture
- Development of new computing platforms (SoCs) devoted for space high-performance computing e.g. Gaisler GR765, nanoXplore ULTRA
- **Growing demand for space HPC applications** (e.g. telecommunication applications)
- Phoenix-RTOS using microkernel architecture widely implemented in IoT applications - good basis for further development

Project motivation – NASA HPSC (High Performance Spaceflight Computing)

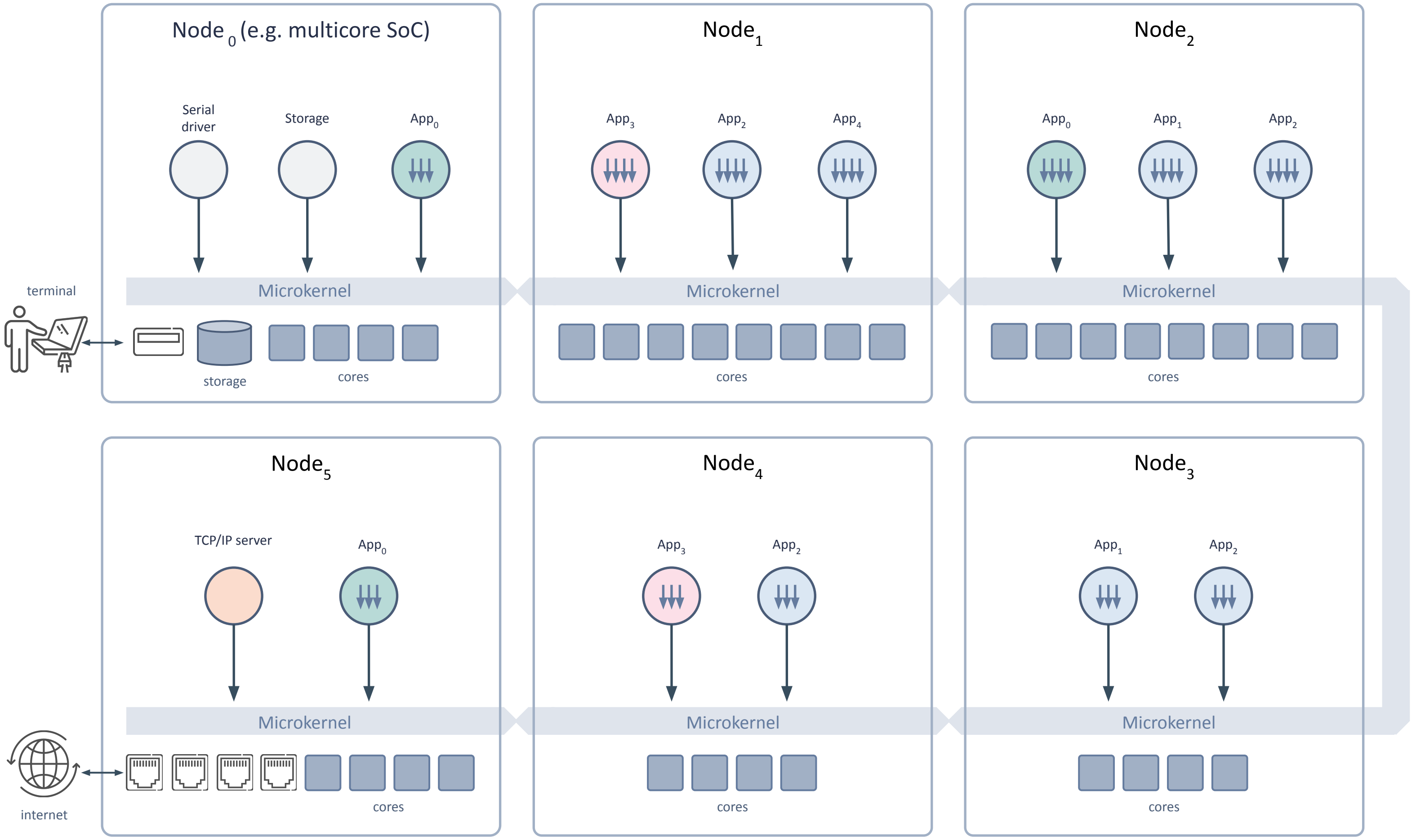
- The goal of the HPSC program is to dramatically advance the state of the art for spaceflight computing
- HPSC will provide a nearly two orders-of-magnitude improvement above the current state of the art for spaceflight processors, while also providing an unprecedented flexibility to tailor performance, power consumption, and fault tolerance to meet widely varying mission needs
- These advancements will provide game changing improvements in computing performance, power efficiency, and flexibility, which will significantly improve the onboard processing capabilities of future NASA and Air Force space missions
- HPSC cost-plus fixed-fee contract was awarded to Boeing. Boeing will provide:
  - Prototype radiation hardened multi-core computing processors (Chiplets), both as bare die and as packaged parts
  - Prototype system software which will operate on the Chiplets
  - Evaluation boards to allow Chiplet test and characterization
  - Chiplet emulators to enable early software development

Space HPC use-cases (identified thanks to NASA HPSC)

Human Spaceflight (HEOMD) Use Cases	Science Mission (SMD) Use Cases
Cloud Services	Extreme Terrain Landing
Advanced Vehicle Health Management Crew Knowledge Augmentation Systems	Proximity Operations / Formation Flying Fast Traverse
Improved Displays and Controls Augmented Reality for Recognition and Cataloging	New Surface Mobility Methods Imaging Spectrometers
Tele-Presence Autonomous & Tele-Robotic Construction	**Radar** Low Latency Products for Disaster Response
Automated Guidance, Navigation, and Control (GNC) Human Movement Assist	Space Weather Science Event Detection and Response
	Immersive Environments for Science Ops / Outreach

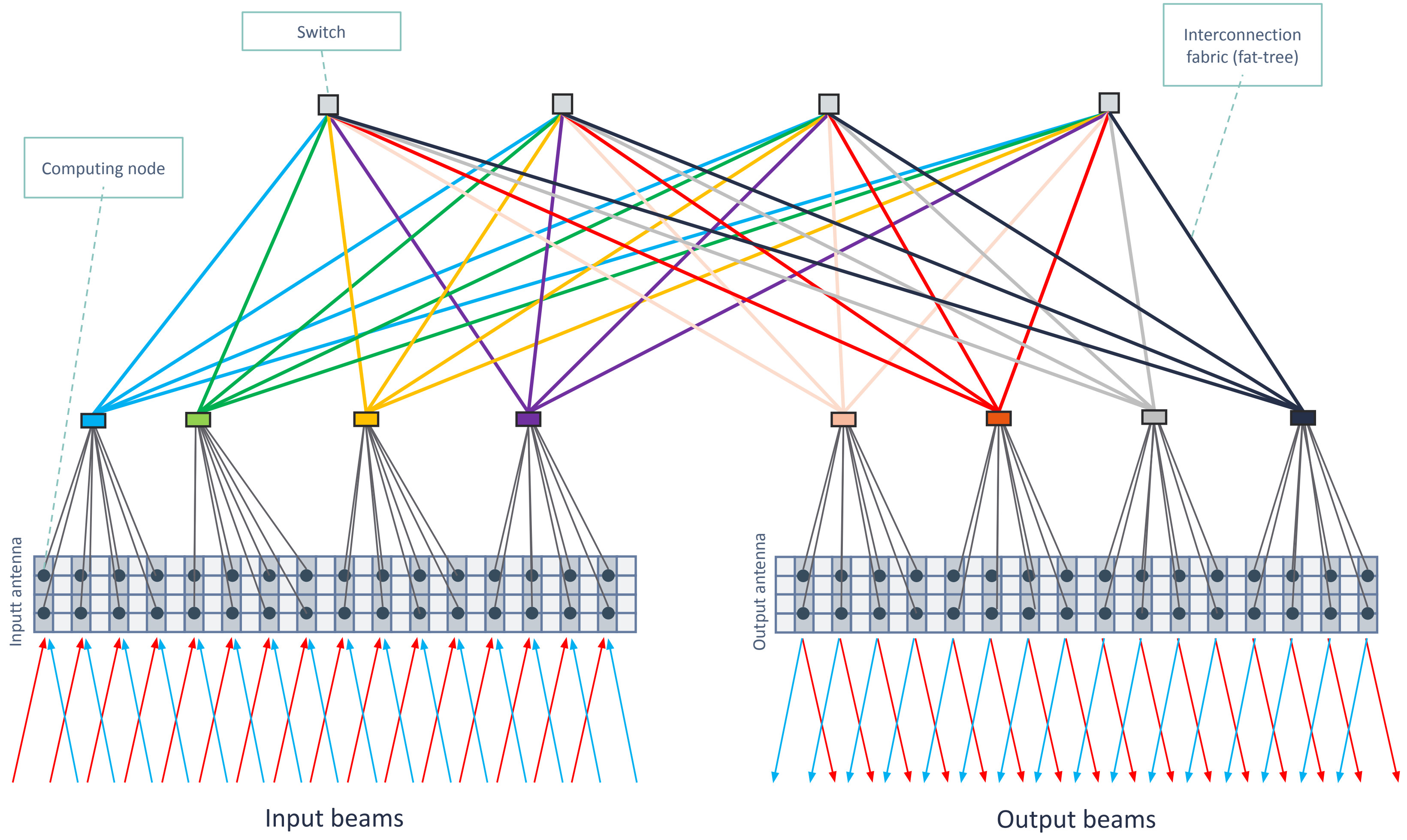
NASA

Distributed operating system - architecture



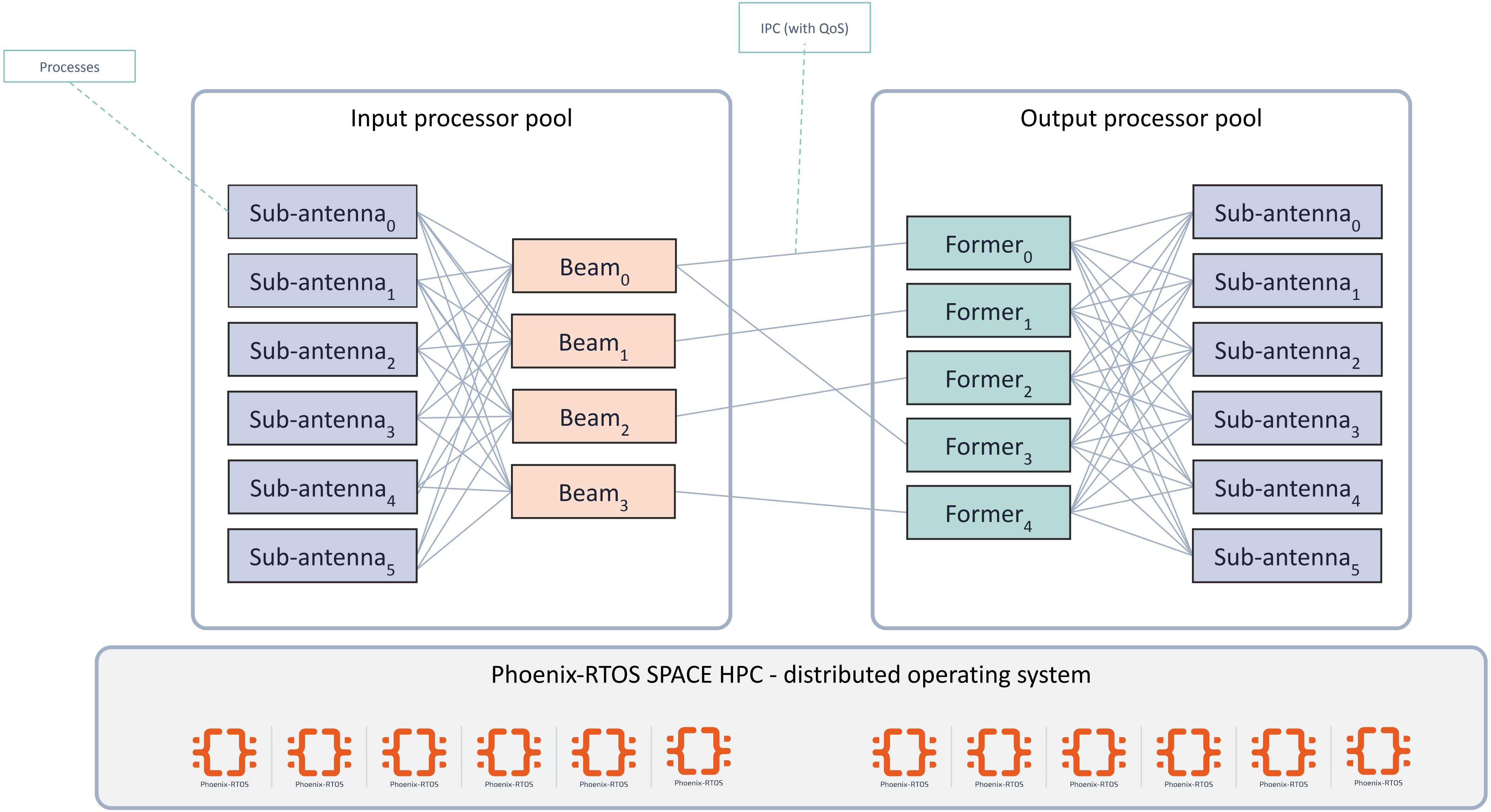
- Computing nodes consisting of processing cores, memory and peripheral devices are connected using efficient interconnection network
- Each node is controlled by microkernel managing memory, processes and threads and providing inter-process communication (IPC)
- Microkernels form virtual bus between components
- Interaction between system components is implemented using message passing and form of RPC (remote procedure call)
- System interacts with external world using using terminal connected to $Node_0$ or Internet connected to $Node_3$

Example application – beam forming digital processor (hardware)



- Using non-blocking interconnect and distributed Phoenix-RTOS, digital data processor for beam forming can be implemented as MPP
- Memories storing samples are assigned to computing nodes connected to sub-antennas
- Uplink and downlink beam forming can be implemented as parallel computation algorithm
- Software-defined solution with high modularity and scalability

Example application – beam forming digital processor (software)



TStorage – NoSQL database for time-series data

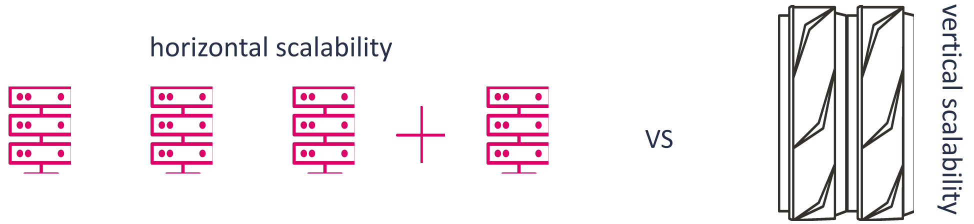
TStorage – NoSQL database

MOTIVATION

- **No existing NoSQL databases for time-series (IoT) available in the open-source model**
- Development of Big Data cloud platforms for smart grid (e.g., energy communities)
- **Over 13 years of experience in the development and maintenance of the object-oriented file system used in multimedia repository**

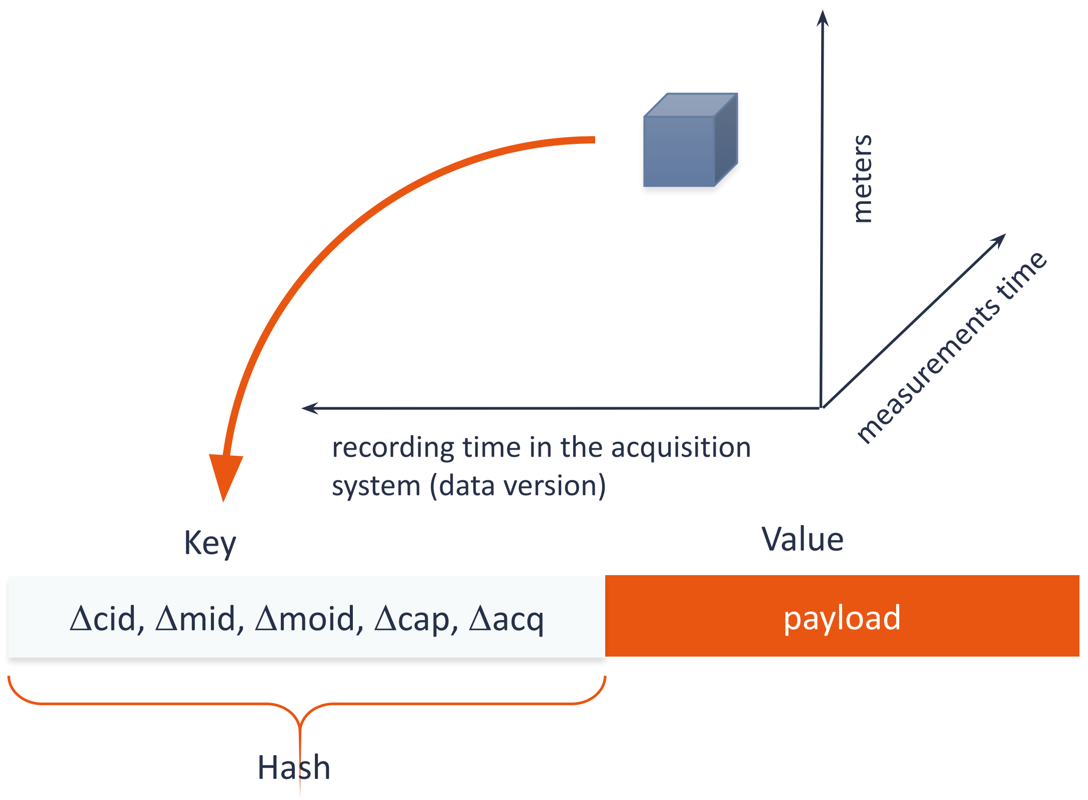
GOAL

- **Horizontal scalability (equipment cost optimization)**
- Quick access to data streams
- Strong semantics of data access suitable for the measurements data
- **High reliability of data storage**



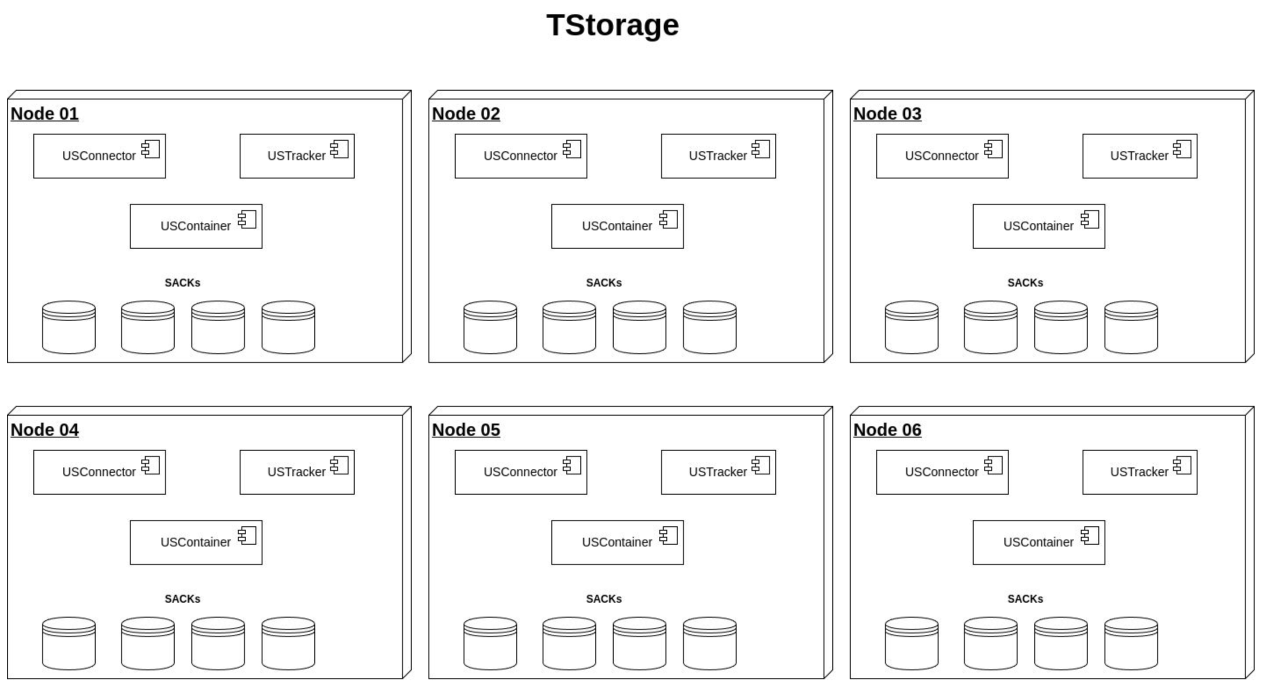
TStorage – data model

- Data model is tailor-made for storing and querying time series data from large networks of IoT devices or sensors
- **5-dimensional database key** allows for performing fast range queries and can be easily mapped on the useful data dimensions, like business domain, metering device, measurement type, measurement time, and database acquisition time
- Native support for data versioning based on acquisition time

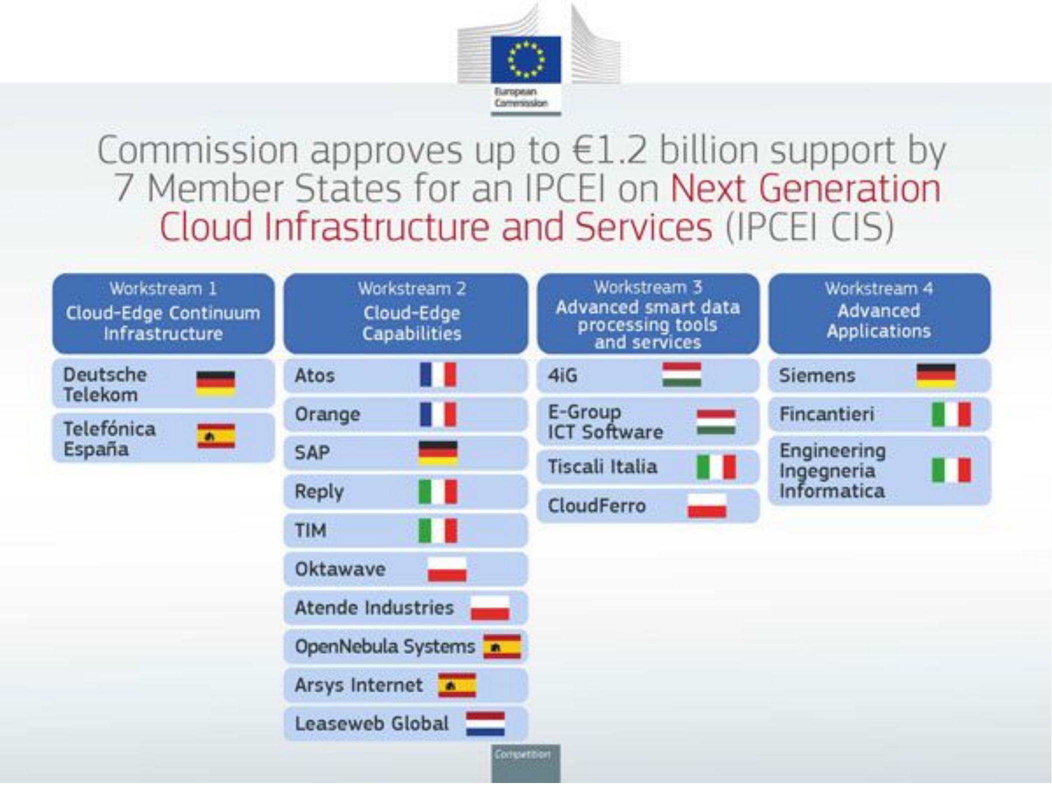


TStorage – distributed database

- Data are distributed between many nodes into many data partitions, enabling system's horizontal scalability.
- As a distributed database, TStorage guarantees global eventual consistency of data and has mechanisms that guarantees strong data consistency for batch processing tasks.
- Native support for data replication between data centers
- Lightweight architecture with small number of external dependencies



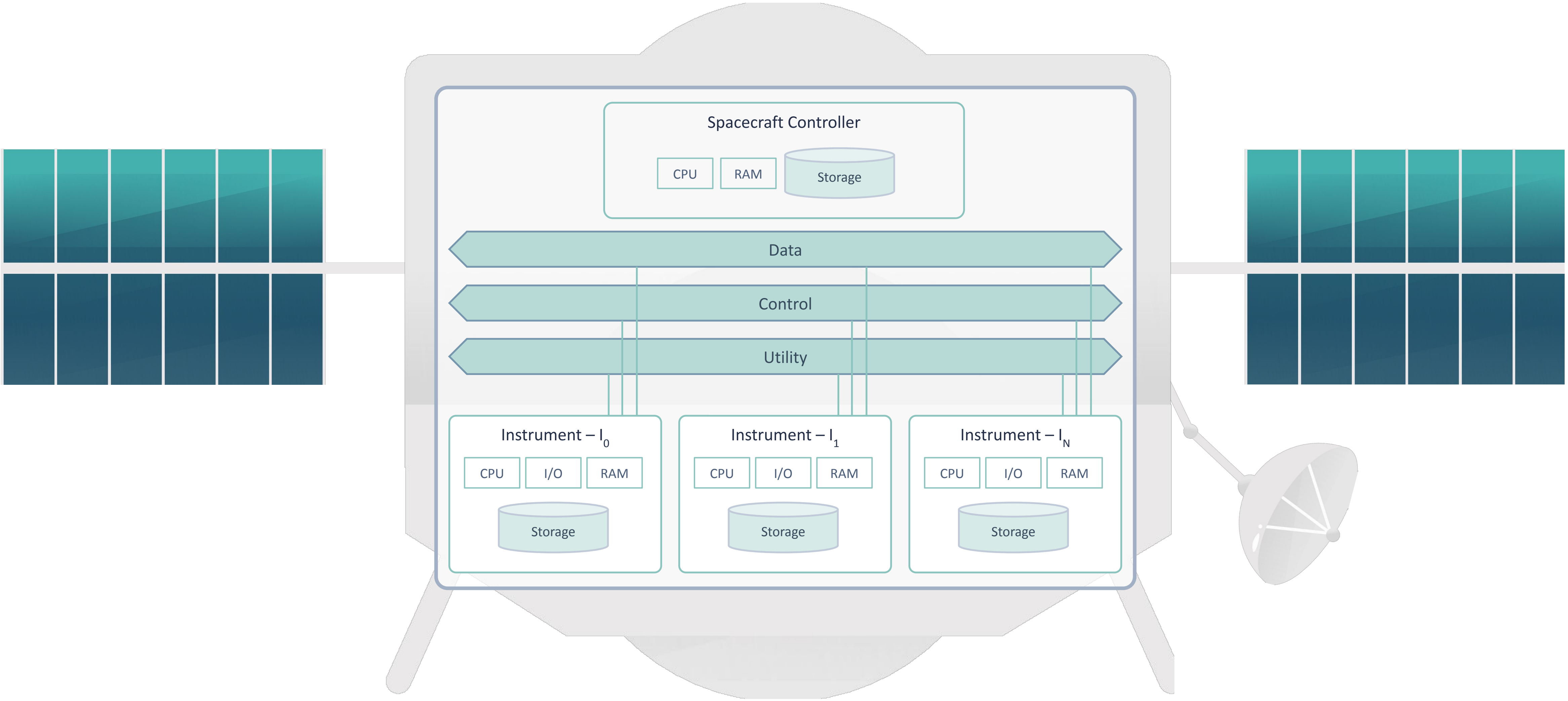
IPCEI-CIS (Important Projects of Common EU interest)



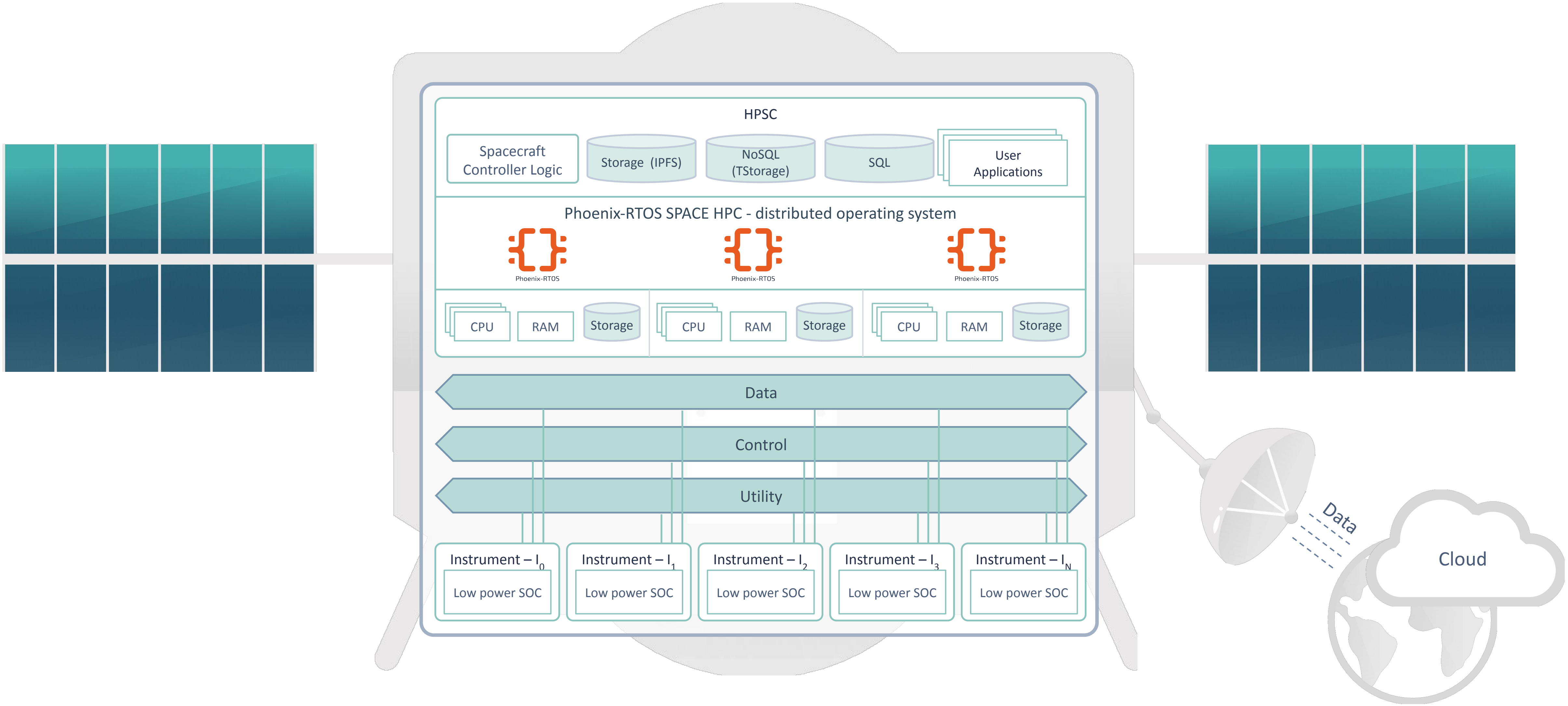
TStorage – use cases for space

- **Ground segment**
  - scalable, distributed and high-performance Big Data storage for time-stamped data from space observation
- **Space segment**
  - NoSQL database for High Performance Spacecraft Computing systems enabling on-board (space-edge) data processing
  - NoSQL database for space data centers

Spacecraft Architecture - current



Spacecraft Architecture - proposed



Phoenix Systems

developer of Phoenix-RTOS

phoenix-rtos.com | github.com/phoenix-rtos

contact@phoenix-rtos.com

POLAND (HQ)
Ostrobramska 86
04-163 Warsaw, Poland

POLAND
Sienkiewicza 10/17
18-400 Łomża, Poland

UNITED KINGDOM
Engine Shed, Station Approach
Temple Meads, Bristol, BS1 6QH, UK