**A Collection of White Papers from the BDEC2 Workshop in Poznań, Poland**

May 14–16, 2019

**Acknowledgment**

This workshop was supported in part by the National Science Foundation under Grant No. 1849625.

# Big Data and Extreme Scale Computing, 2nd Series, (BDEC2)
## Workshop 3: Poznan, Poland,
## May 14-16, 2019

# Call for White Papers

Previous BDEC editions have highlighted a consensus on the need for a software platform of distributed services enabling science-driven complex workflows across a continuum of edge and shared centralized (e.g. HPC, Cloud) resources, together with their data logistics.

These evolutions not only question the software convergence of HPC/HDA at a technological level but also the ability to provide world-wide compatible and flexible solutions / services that take into account the pervasive aspects of data and the data logistics in a heterogeneous, distributed environment. This environment must combine Edge, Cloud and/or HPC infrastructures to support science-driven applications and innovative scientific discovery processes that increasingly include Machine Learning (ML) and Deep Machine Learning (DML) techniques.

Computational and data resources governance, access, allocation, management, together with innovative execution protocols and programming environments, cannot be dismissed from the BDEC picture, as many of the major scientific initiatives (e.g. Coupled Model Intercomparison Project of the IPCC, Square Kilometer Array Project, Large Hadron Collider, Space Earth Observations, etc.) involve international teams that are sharing and processing data produced from large, centralized or decentralized, scientific instruments and observational systems.

## Platforms

On one hand we are looking for a kind of ability to provide technical solutions (e.g. global data set and resource registries) that are consistent and interoperable between the major infrastructures, and on the other hand to be agile enough (e.g. openstoragenetwork NSF initiative) in order to adapt to the fast-changing landscape.

The Poznan BDEC workshop aims to bring together application domains, research informatics, cyber-infrastructure application developers and providers to focus research and technological thinking on the disruptive impact of:

1) Large scale, science-driven workflows — orchestrating high-end data analysis, intensive computing stages, with increasing use of AI techniques – across a compute- and data continuum of edge and shared centralized resources (2nd area of the picture below).
2) Data logistics all along these workflows and across the compute- and data continuum.

Our objective is to identify potential driving *demonstrators* (see below) able to illustrate a set of issues and solutions being faced by the community, together with identifying possible

convergence between these, and to co-develop and co-design a community-driven shaping strategy.

For this workshop we welcome white papers addressing these 2 topics. To allow the emergence of a grand picture and leverage a common understanding from this meeting, we would like all white papers to follow a set of analysis criteria:

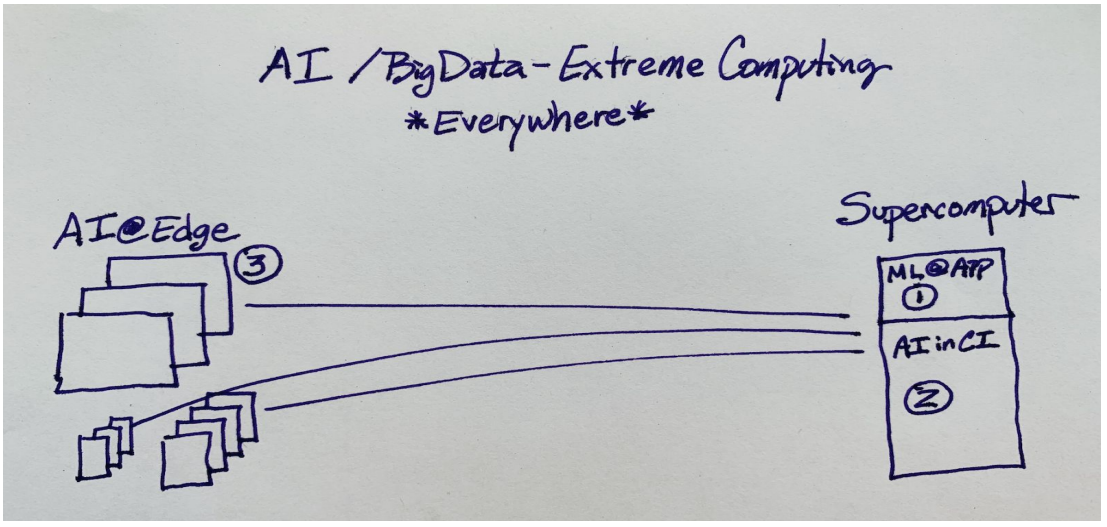
a) What new "services" should be available to support current and future science-driven workflow applications and their data logistics?
b) What technologies could be used to enable and implement these new services?
c) How to co-design a community-driven shaping strategy to define and develop these services, and make these technologies widely available, accepted, and supported across a continuum of shared infrastructures?

## Demonstrators

As mentioned above, the objective of the BDEC2 project is to organize and co-develop an international community-driven shaping strategy *demonstrating* the feasibility and the potential of a new software platform of distributed services that will enable applications across edge and shared centralized infrastructures, together with their data logistics. The time frame for delivery of these demonstrators is expected to be Q4 2020.

The Poznan BDEC2 meeting will start to bring together research thinking and technological developments within a community-driven shaping strategy. Consequently, we also welcome white papers presenting demonstrator *candidates*, that highlight one or more of these elements:

- BDEC2 DCP (distributed computing platform) driven by desirable/aspirational, *disruptive* demonstrators (either in terms of capability or science) that cannot (practically speaking) be done now.
- Community-driven shaping strategy for enabling science-driven, large-scale workflows involving *AI* techniques, together with their data logistics, across a continuum of edge and shared centralized infrastructures.



# Towards a demonstrator of the Sigma Data Processing Architecture for BDEC 2

Gabriel Antoniu[1], Alexandru Costan[1], Ovidiu Marcu[1], Maria S. Pérez[2], Nenad Stojanovic[3]

[1]Univ Rennes, Inria, CNRS, IRISA

[2]Universidad Politécnica de Madrid

[3]Nissatech

1 May 2019

At the first BDEC 2 workshop held in Bloomington we presented a white paper introducing the Sigma data processing architecture, which aims to enable unified data processing across hybrid infrastructures combining edge, cloud and HPC systems. To do so, it aims to relevantly leverage and combine stream processing and batch processing in situ and in transit. We now introduce several software libraries and components based on which the Sigma architecture can be implemented.

**Our vision in a nutshell.** Due to an ever-growing digitalization of the everyday life, massive amounts of data start to be accumulated, providing larger and larger volumes of historical data (**past data**) on more and more monitored systems. At the same time, an up-to-date vision of the actual status of these systems is offered by the increasing number of sources of real-time data (**present data**). Today's data analytics systems correlate these two types of data (past and present) to predict the future evolution of the systems to enable decision making. However, the relevance of such decisions is limited by the knowledge already accumulated in the past. Our vision consists in improving this decision process by enriching the knowledge acquired based on **past data** with what could be called **future data** generated by simulations of the system behavior under various hypothetical conditions that have not been met in the past. This can enable hybrid modeling combining simulation models (running on HPC systems) and data-driven models (running on clouds or on cloud+edge infrastructures), to enable higher-precision data analytics (**Hybrid Analytics**). To support such analytics we advocate for unified data processing on converged, extreme-scale distributed environments thanks to a novel data processing architecture able to relevantly leverage and combine stream processing and batch processing in situ and in transit: **the Sigma architecture for data processing**.

**Reminder of the Sigma Architecture for Data Processing.** Traditional *data-driven analytics* relies on *Big Data processing* techniques, consisting of *batch processing* and *real-time (stream) processing*, potentially combined using for instance the *Lambda architecture*. This architecture uses batch processing to provide comprehensive and accurate views of batch data, while simultaneously using real-time stream processing to provide views of online data.

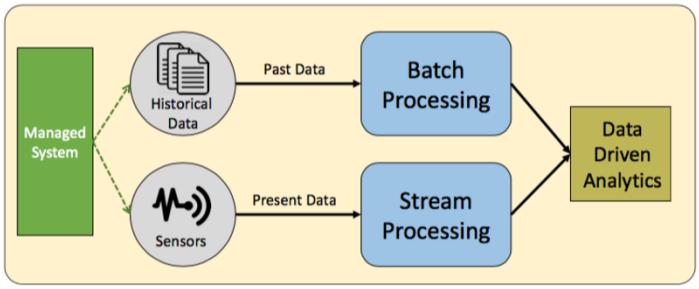



*Figure 1. The Lambda data processing architecture.*

On the other side, *simulation-driven analytics* is based on computational (usually physics-based) simulations of complex phenomena, which often leverage HPC infrastructures. The need to get fast and relevant insights from massive amounts of data generated by extreme-scale simulations led to the emergence of *in situ* and *in transit* processing approaches [Bennet2012]: they allow data to be visualized and processed interactively in real-time as data are produced, while the simulation is running.

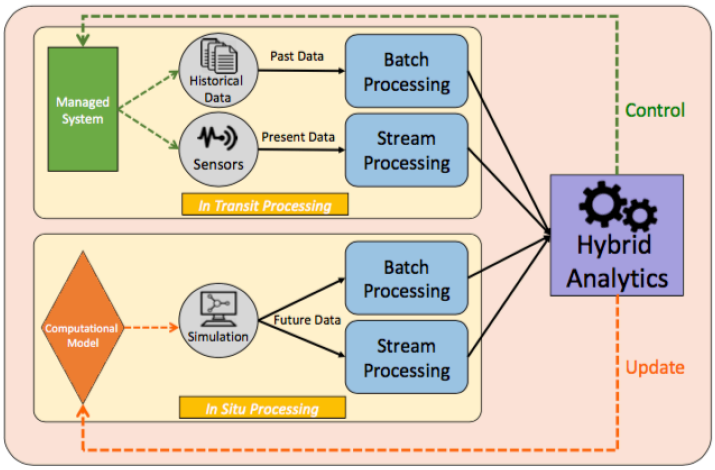



*Figure 2. The Sigma data processing architecture.*

To support hybrid analytics and continuous model improvement, we propose to combine the above data processing techniques in what we will call the **Sigma architecture**. It combines batch-based and stream-based Big Data processing techniques (i.e., the Lambda architecture) with in situ/in transit data processing techniques inspired by the HPC (Figure 2). This allows to collect, manage and process extreme volumes of past, real-time and simulated data. The architecture relies on two layers:

- The *data-driven layer* leverages the unified Lambda architecture to *update the simulation model* dynamically using the past and real-time data through a continuous learning loop [Marcu2017].
- The *computation-driven layer* uses in situ and in transit processing to *proactively control in real-time* the targeted systems. To enable this capability, in situ processing platforms need to be enhanced with Big Data analytics support. This will lead to a very efficient management of the targeted systems, based on new sets of services, not existing in current systems, like: *proactive controlling* (e.g., real-time response to unknown anomalies) based on performing data-analytics on past, present and future data; *on-demand computational simulations triggered in real-time*, driven by data analytics, in order to find opportunities for improvement, tuning and for learning forecasting system models from past data. This is expected to reduce uncertainty in prediction and to improve decision making.

To provide a reference implementation of the Sigma architecture, our approach consists in jointly leveraging two existing software components: the Damaris middleware for scalable in situ/in transit processing and the KerA unified system for ingestion and storage of data for scalable stream processing.

**Damaris: scalable in situ and in transit processing on HPC systems.** Damaris is a middleware for scalable I/O management and in situ/in transit visualization on large-scale HPC systems. It relies on dedicated cores/nodes for asynchronous I/O or in situ/in transit processing. Developed at Inria in the framework of a collaboration within the JLESC international lab, it scaled up to 16,000 cores on Oak Ridge's leadership supercomputer Titan (first in the Top500 supercomputer list at the time of the experiments, in 2013) before being validated on other top supercomputers. Active development is currently continuing at Inria, where it serves as a basis for several strategic emerging collaborations with industry (e.g. Total).

**KerA: unified ingestion and storage for low-latency, high-throughput stream processing on clouds and cloud+edge infrastructures.** KerA is a unified architecture for stream ingestion and storage aiming to optimize data processing in Big Data applications (low-latency and high throughput). It minimizes data movement within the analytics architecture, finally leading to better utilized resources. Its design principles include: a unified data model for objects and streams; eliminating data redundancies between ingestion and storage; dynamic partitioning for flexible and elastic management of stream partitions. We implemented and preliminarily evaluated a software prototype for KerA with the goal of illustrating its efficient handling of diverse access patterns: low-latency access to streams and/or high throughput access to unbounded streams and/or objects.

**Damaris+KerA: towards a framework for scalable in situ/in transit analytics on hybrid infrastructures (HPC+cloud+edge).** Damaris has already been integrated with HPC storage systems (e.g., HDF5) to enable scalable collective writing of data generated by simulations running on tens of thousands of cores (e.g., at the end of each iteration). Such data can further be analyzed (typically through offline analysis). We developed for Damaris a new real-time storage backend based on KerA: leveraging Damaris dedicated cores and shared memory

components, we are able to asynchronously write in real-time the data output of a simulation (that corresponds to multiple iterations) as a series of events, logically aggregated by a stream, with metadata describing the stream's sub-partitions for each iteration. This is an important step towards a complete data processing and analytics workflow: by leveraging real-time streaming analytics (e.g., Apache Flink) efficiently coupled with KerA through shared memory, scientists can further apply machine learning techniques *in real time, in parallel with the running simulation*.

**Questions and Answers**

1. *What innovative capabilities/functionalities will the proposed candidate platform demonstrate (e.g. transcontinuum workflow, edge computing, data logistics, distributed reduction engine, etc.)?*

The above integration illustrates how data processing techniques and supporting technologies from the HPC area (in situ/in transit processing) and Big Data worlds (stream processing) can be composed in hybrid scenarios mixing HPC simulations and Big Data analytics. It provides the possibility to explore many scenarios combining simulations and analytics, for instance, to launch other simulations on demand, in parallel with the initial simulation (e.g., based on the output of a previous iteration of a simulation) in order to explore various models. As a further step, we are investigating solutions to extend this approach towards edge infrastructures, to support even more hybrid workflows running on hybrid HPC/Cloud/edge infrastructures.

2. *What applications/communities would/could be addressed?*

Applications exhibiting scenarios that mix simulations and analytics in parallel are concerned.

3. *What is the "platform vision," i.e. what kind of shared cyberinfrastructure (CI) for science would the further research/design/development of this platform lead to?*

A converged HPC+cloud+edge infrastructure supporting workflows that combine simulations and real-time analytics.

4. *How available/ready/complete is the set of software components to be used to build the demonstrator?*

Damaris is distributed as an open-source library [Damaris]. KerA is operational and has been preliminarily evaluated on cluster/cloud testbeds, it is not open-sourced. We are working on an integrated platform based on KerA and Damaris to extend the KerA+Damaris integrated data processing framework to support edge infrastructures (work in progress).

5. *As far as one can tell at this early date, to what extent can this be done with existing and/or otherwise available hardware/software/human resources?*

Building a full, convincing demonstrator requires identification of concrete use cases, (in particular, from industry) joint international collaboration and close interaction with the corresponding partners. We believe substantial extra human resources and dedicated efforts are required.

6. *What is the potential international footprint of the demonstrator?*

High. Both KerA and Damaris have already been developed through international efforts. The Sigma architecture is the result of an international design effort (as illustrated by the list of authors of this document). We plan to jointly submit H2020 project proposals to continue such joint international efforts. We are open to international collaborations within BDEC, enabling integration/connection to other platforms in global, converged scenarios. We are looking for use cases at the international scale. We are also in the process of building a startup that aims to leverage the Sigma architecture.

**References**

[Ibanez2017] R. Ibanez, E. Abisset-Chavanne, J.V. Aguado, D. Gonzalez, E. Cueto, F. Chinesta. A Manifold-Based Methodological Approach to Data-Driven Computational Elasticity and Inelasticity. Archives of Computational Methods in Engineering, 2017.

[Bennet2012] J.C. Bennet, H. Abbasi, P.-T. Bremer, R. Grout et al. Combining in-situ and in-transit processing to enable extreme-scale scientific analysis. In Proc. ACM SC'12, Salt Lake City, Nov. 2012.

[Carbone2015] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, K. Tzoumas, Apache Flink: Stream and batch processing in a single engine, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 36 (4).

[Damaris] The Damaris project. https://project.inria.fr/damaris/

[Dorier2012] M. Dorier, G. Antoniu, F. Cappello, M. Snir, L. Orf. "Damaris: How to Efficiently Leverage Multicore Parallelism to Achieve Scalable, Jitter-free I/O", In Proc. CLUSTER – IEEE International Conference on Cluster Computing, Sep 2012, Beijing, China. URL: https://hal.inria.fr/hal-00715252

[Dorier 2013] M. Dorier, R. Sisneros, T. Peterka, G. Antoniu, D. Semeraro, "Damaris/Viz: a Nonintrusive, Adaptable and User-Friendly In Situ Visualization Framework", Proc. LDAV – IEEE Symposium on Large-Scale Data Analysis and Visualization, Oct 2013, Atlanta, USA. URL: https://hal.inria.fr/hal-00859603

[Marcu2017] O. Marcu, A. Costan, G. Antoniu, M.S. Pérez, R. Tudoran, S. Bortoli and B. Nicolae, Towards a Unified Storage and Ingestion Architecture for Stream Processing. IEEE International Conference on Big Data (Big Data), 2402–2407; 2017. DOI: 10.1109/BigData.2017.8258196

# Personalized healthcare: workflow orchestration for cyberinfrastructure platforms

Rosa M Badia, Miguel Vázquez, Sergi Girona
Barcelona Supercomputing Center

## 1. What innovative capabilities/functionalities will the proposed candidate platform demonstrate (e.g. transcontinuum workflow, edge computing, data logistics, distributed reduction engine, etc.)?

This demonstrator will focus on the orchestration of workflows across the platform. The demonstrator will be based on COMPSs, which while it is a task-based programming model it also provides means for the development and orchestration of edge-to-cloud/HPC workflows. The COMPSs runtime has been recently modified to become a decentralized engine to fit the requirements of such infrastructures. For such scenario, COMPSs is combined with dataClay which offers an object-oriented library that is able to federate in-memory data from multiple stores.

These are the main innovations on the demonstrator:

- Orchestration of dynamic end-to-end workflows in an edge-to-cloud/HPC platform, that take into account the volatility of the edge devices. By dynamic, we mean that the actual workflow depends on the actual inputs of the workflow and also that can change at execution time due to the occurrence of specific events.
- Integration of machine learning, data analytics and computational processes in a single workflow.
- Multiple aspects of data management in such infrastructures: sharing, privacy, locality, security, management of devices' failures or disappearance, etc.

## 2. What applications/communities would/could be addressed?

For the BDEC demonstrator we want to propose a use case based on **personalized healthcare**, with wearable devices collecting continuous information on multiple parameters of the patient. The parameters will be processed in the edge and used to train AI models in the cloud/HPC to provide personalized notifications, alerts, and recommendations for prevention. To avoid personal data to leave the controlled environment of edge computing, data reaching the cloud will be encrypted or encoded.

However, since COMPSs is a general-purpose programming model, there is a wide range of applications where what we are proposing can be applied. We are contributing to multiple European funded projects involving fog-to-cloud scenarios where the COMPSs programming models is used. Find here some examples:

- mF2C: smart airport fog hub, enriched boat navigation service, emergency management in smart cities
- CLASS: smart city use, with heavy sensorized urban area and connected cars

- ELASTIC: smart tramway

## 3. What is the "platform vision," i.e. what kind of shared cyberinfrastructure (CI) for science would the further research/design/development of this platform lead to?

We are considering a hierarchical platform with devices in the edge (wearable devices) that are the sources of data, mobile devices in the edge connected to the wearable devices that can be used for prediction and cloud/HPC devices used to train the models.

## 4. How available/ready/complete is the set of software components to be used to build the demonstrator?

COMPSs runtime is already functional in fog-to-cloud scenarios. The use of HPC instead of cloud has not been tried, but should be almost immediate.

The actual personalized healthcare use case is at proposal level, therefore needs to be developed.

## 5. As far as one can tell at this early date, to what extent can this be done with existing and/or otherwise available hardware/software/human resources?

The wearable devices are not available right now, and some of the development resources are not allocated. The cloud/HPC devices could be provided by BSC.

## 6. What is the potential international footprint of the demonstrator?

As we have said before, the COMPSs runtime is already involved in multiple scenarios that involve several European countries. Given the nature of the use case that we are considering, the demonstrator has a potential for becoming very attractive and we expect contributors from worldwide level.

# BDEC Platform Demonstrator: A Global Data Logistics Platform for the Digital Contiuum

Micah Beck, Terry Moore, Piotr Luszczek
University of Tennessee
Ezra Kissel, Martin Swany
Indiana University

**Introduction —** People in the BDEC community are well positioned to understand the obstacles to research cooperation that different kinds of boundaries—national, physical, social, organizational, and technological—can produce. From years of experience building cyberinfrastructure for science and engineering, we know that, inevitably, the individuals who make up the international scientific community are embedded in a network of complex, multi-dimensional relationships that constrain their freedom to control or share their computing and data resources with potential collaborators across these boundaries. We also know that for the past three decades, the effects of such borders were mitigated and partly overcome by the TCP/IP+Unix/Linux platform paradigm, whose near universal adoption and use provided the foundation for interoperability and software portability throughout the community. But as described in the BDEC Pathways to Convergence report [1], this paradigm is becoming progressively more inadequate: swamped by the ongoing data tsunami; overwhelmed by swarms of new digital devices proliferating in the surrounding environment; and, most of all, rendered increasingly irrelevant and impervious to innovation for the purposes of science by its subjugation to the private ends of a few global cloud service corporations. Against this background, the BDEC demonstrator described below—*a Data Logistics platform (DLP ) for the digital continuum*—expresses our conviction that the way forward to a next generation cyberinfrastructure for science is through a radically new platform design, one that meets the application challenges of the our transformed world while still achieving the same community-wide levels of adoption and use (of "deployment scalability") that the legacy paradigm enjoyed.

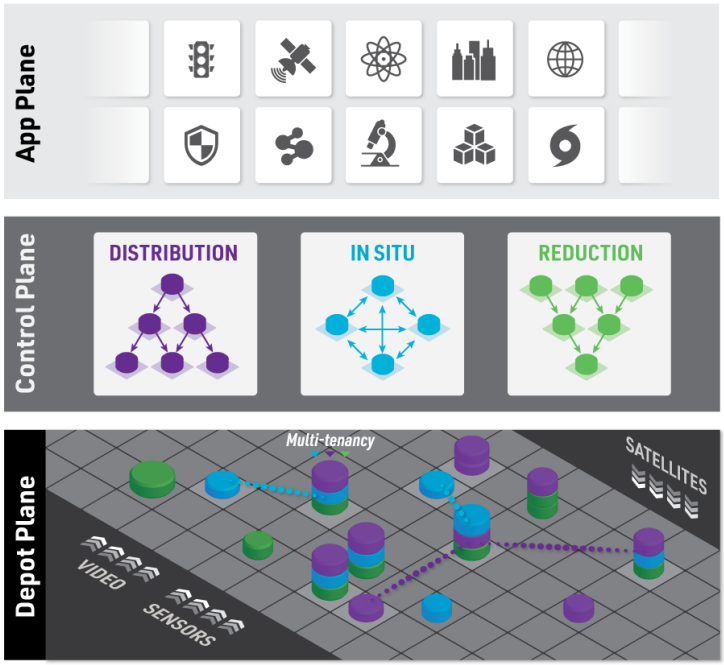



**Figure 1:** Simplified view of the *exposed buffer processing* (EBP) architecture for the DLP demonstrator: the local node layer composed of passive resource nodes (called *depots*) accessed via EBP protocols; the control layer with diverse local area or global service managers (e.g., file system, CDN, in transit/in situ data processing, etc.); and the application layer, where applications use the control plane managers to perform work on data in the node plane.

**Vision of a Data Logistics platform Spanning the Digital Continuum—** A data logistics platform is a distributed system in which the system's intermediate nodes, called *depots*, are used in a general way. That is, they don't just forward packets, but instead expose all three fundamental resources—communication, storage/buffer, and processing—to explicit programming. The BDEC DLP we propose to demonstrate is based on the idea that, in order to achieve the continuum-spanning interoperability and portability we seek, we have to build on a common service virtualization, or *spanning layer*, that is designed to be as simple, general and limited as possible while still supporting all necessary applications [2]. These requirements are fulfilled in the specification of a converged service that represents the "greatest common divisor" of storage, networking and computation, namely, *the allocation of, transfer of data between, and transformation of data in* ***buffers***, including both memory and storage and regardless of implementation. Accordingly refer to this

design as an *exposed buffer processing* (EBP) architecture [3].

Figure 1 provides an overview its basic structure of this architecture and the BDEC DLP demonstrator. Using the Data Logistics Toolkit (*DLT*, [4]), it will be build on a collection of *DLT* depots deployed at a set of locations throughout the global research network. These depots, which constitute the data plane of the DLP give clients the ability to perform a set of low level fundamental operations, to whit: **1)** allocate a buffer, **2)** move data between buffers on a single or between LAN-connected depots or **3**) apply an operation to a set of buffers that serve the roles of inputs and/or outputs according to the definition of the operation. The protocol and service model are uniform across all depots in the DLP , whether they are providing access to the resources of nodes located at the very edge, the middle (or core) or at a data center. Thus access to the resources of all nodes that comprise the computing continuum are represented with as much uniformity and interoperability as possible. Nodes with greatly varying attributes may be differentiated by the publication of appropriate metadata (for instance distinguishing a node resource that exposes line buffers for transient buffer of network transfers from one that provides long lived storage cells for implementing file abstractions). But DLP data transfer must be interoperable between any two nodes that are adjacent in the LAN topology, with no intervening translators or gateways. Routing along paths is within the architecture, but gateways to overcome non-interoperability are not. Operations on data may similarly be implemented on some nodes, as described in appropriate metadata, but the same operation on two different nodes must be semantically indistinguishable.

Higher level services will be implemented by aggregating the fundamental operations that the depots support. These services are implemented by processes running in a control plane that has access to the entire data plane that models the computing continuum. In simple cases an entire service can be created by running a single process that issues commands to the nodes of the control plane, centralizing the logic of the service while distributing its execution to the data plane. In cases where such extreme physical centralization of the control plane causes problems (typically performance or reliability) a single process may be implemented by a set of processes that communicate using both the depots of the data plane and the Internet.

Having deployed a data plane, a particular (set of) demonstration application(s) will be chosen and a set of control plane services defined that are sufficient to implement it. A natural choice would be a prototype of the Earth Observing Data Network, a generalized Content Distribution Network that distributes a stream of satellite images emanating from one or a small set of orbiting instruments. The generalized nature of EODN allows it to support not only timely high performance access to satellite data in real time, but also allows light processing of those images using the nodes of the data plane as well as download to client facilitites. EODN also supports the upload by clients of secondary data products into EODN for further processing either in the data plane or after download by other clients. No such general, open, *cooperatively managed and operated* CDN currently exists for the global distribution of filtered, location-specific data from one nation's satellite, let alone a CDN for the distribution of relevant data from the satellites of international partners. The appetite for such data around the world, both inside and outside the scientific community, is substantial.

**The Data Logistics Toolkit (*DLT*) as the Basis of the BDEC DLP** The data storage and transfer functions of the DLP data plane will be implemented by the Internet Backplane Protocol (*IBP* depot, [5] ), which implements the EBP spanning layer as an overlay on the legacy platform. A production quality, packaged implementation of the *IBP* depot is available in the NSF funded Data Logistics Toolkit (*DLT*). Light computing on data stored in *IBP* depots have been implemented experimentally, but this function is not currently part of the DLT. It will have to be reimplemented. The data storage, distribution and access functions of EODN are supported by the Intelligent Data Movement Service (IDMS) utility, and the IDMS policy can be controlled by the Flange languageṪhese capabilities are included in the current DLT distribution. Light processing of data within the data plane will require the augmentation of both IDMS and Flange with appropriate control functionality. Access to IDMS from an appropriate end-user interface, commands or client API will also be also required, as well as a means to harvest the stream of satellite images in real time.

**Innovations of the BDEC DLP —** The main innovations of the DLP derive from its architecture. The key one—a spanning layer that provides buffer-based interoperability and portability across the entire continuum—has been highlighted above. Some other unique attributes are briefly discussed below. It should be noted, however, that since the BDEC demonstrator will be implemented as an overlay on the legacy paradigm, it inherits the liabilities of that paradigm, specifically as regards security. The benefits of "Inherent support for role based, federated security" described below will have to await a native, or non-overlay implementation; the need for a new way to address such issues with the legacy paradigm is one the primary motives finding path to transition away from it. The BDEC DLP demonstrator will open that path.

- *Exposed topology for effective data logistics:* Approaches that hide topology from their clients are inherently inadequate platforms on which to build high traffic, globally distributed systems. To achieve efficient and performant data logistics, the EBP spanning layer exposes topology, including the placement and allocation of bandwidth, storage/memory and processing across the system. However, such exposed topology is generally too volatile to be "source scheduled" at one place in the network. A well tested solution is to create tightly controlled subdomains (i.e., Autonomous Systems (AS) and subnets) that peer at their boundaries. Defining subdomains allows for aggregate characterizations of platform topology that permit the level of exposure to be varied as the situation requires.
- *Decentralized service model enables local autonomy:* The creation of peering subdomains that control their own node layer resources has the added advantage decentralizing administration, giving localities the freedom of manage how their resources are shared. Since higher level functions do not need to have the same deployment scalability as the depot plane infrastructure, they may be centralized where necessary. This enables heterogeneity in the creation and management of higher level functions, allowing nodes to be very close (in network topology) to sensors, actuators and other edge devices. This combination of autonomous localities connected together by services based on weak assumptions will enable logistical services that efficiently make use of all the resources of the continuum.
- *Inherent support for role based, federated security:* The DLP spanning layer is local to the depot and doesnt export a global service; services that use it cannot assume the global reachability of any given local node. By minimizing the "target" that the node infrastructure offers to any external input or signal, DLP can create a kind of "white list" of allowable global services, as defined by the privileged control plane, leaving the node as impervious as possible to communication that is not part of such an authorized service. Then each higher level service can define its own strategy for authenticating, protecting and allowing access to the service it creates.

**International Participation and Feasibility of the BDEC DLP Demonstrator —** The fact that depots in the data plane can be deployed and used in a completely decentralized fashion makes participation the most basic form of participation in the DLP demonstrator straightforward: install one or more IBP depots on available hardware and give the result a suitable network connection. As descibed in [4], to achieve such easy deployment and configuration, DLT software has been packaged and documented for a number of operating system distributions; an accompanying meta-package resolves and installs any necessary dependencies needed to deploy an IBP depot node, as well as other companion DLT modules(e.g., Phoebus WAN accelerator, Periscope instrumentation, etc.), and it registers the new depot with the IDMS (see below). Available versions also include appliance images (VMs) that can be deployed on OpenStack- and Emulab-based rack technologies, as well as containerized versions with supporting documentation.

At the level of the control plane, we expect the *DLT*'s IDMS and policy engine to work for the BDEC DLP out of the box. Since the DLP depot plane provides a "bits are bits" infrastructure, the same functionality will be available to other application communities (e.g., microscopy, astronomy, etc.). But as Figure 1 suggests, various other more powerful services can be implemented on the control plane of the DLP , given availability operations installed on the depots. Whether the demonstrator plan is expanded in this way will likely be a function of community interest and available resources.

## References


[1] M Asch, T Moore, R Badia, M Beck, P Beckman, T Bidot, F Bodin, F Cappello, A Choudhary, B de Supinski, E Deelman, J Dongarra, A Dubey, G Fox, H Fu, S Girona, W Gropp, M Heroux, Y Ishikawa, K Keahey, D Keyes, W Kramer, J-F Lavignon, Y Lu, S Matsuoka, B Mohr, D Reed, S Requena, J Saltz, T Schulthess, R Stevens, M Swany, A Szalay, W Tang, G Varoquaux, J-P Vilotte, R Wisniewski, Z Xu, and I Zacharov. Big data and extreme-scale computing: Pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry. *The International Journal of High Performance Computing Applications*, 32(4):435–479, 2018. doi: 10.1177/1094342018778123. URL https://doi.org/10.1177/1094342018778123.

[2] Micah Beck. On the Hourglass Model, End-to-End Arguments, and Deployment Scalability. *Communications of the ACM*, to appear, July 2019.

[3] Micah Beck, Terry Moore, Piotr Luszczek, and Anthony Danalis. Interoperable convergence of storage, networking, and computation. In Kohei Arai and Rahul Bhatia, editors, *Advances in Information and Communication*, pages 667–690, Cham, 2019. Springer International Publishing. ISBN 978-3-030-12385-7.

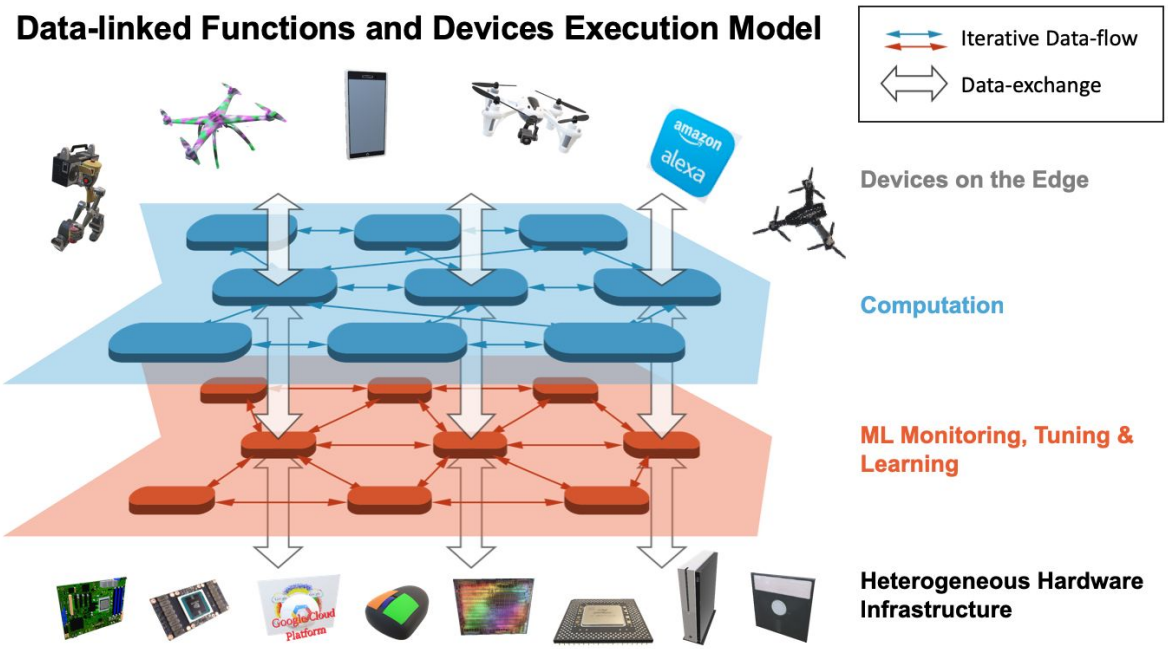
[4] Ezra Kissel, Micah Beck, Nancy French, Martin Swany, and Terry Moore. Data Logistics: Toolkit and Applications. *GOODTECHS 2019 - 5th EAI International Conference on Smart Objects and Technologies for Social Good*, 2019. URL http://bit.ly/DLT-GoodTechs. submitted.

[5] A. Bassi, M. Beck, G. Fagg, T. Moore, J. S. Plank, M. Swany, and R. Wolski. The Internet Backplane Protocol: A study in resource sharing. In *Cluster Computing and the Grid, 2002. 2nd IEEE/ACM International Symposium on*, pages 194–194, May 2002. doi: 10.1109/CCGRID.2002.1017127.

## Twister2 Demonstrations for BDEC2

Geoffrey Fox, Vibhatha Abeykoon, Selahattin Akkas, Kannan Govindarajan, Gurhan Gunduz, Supun Kamburugamuve, Niranda Perera, Ahmet Uyar, Pulasthi Wickramasinghe, Chathura Widanage,
*Digital Science Center, Indiana University Bloomington; gcf@indiana.edu*

We can demonstrate a novel middleware Twister2 [1], [2] that supports programming in components linked together by high performance messaging. This model captures recent cloud (native) ideas such as Function as a Service and microservices; it also supports geographically distributed computing as one gets when linking edge device, fog, and data-center components. One also sees a need for linked components in many familiar HPC scenarios such as those linking different simulations for different parts of a complete system. Recent studies of the use of Machine Learning (ML) to enhance HPC suggest this architecture sketched in the figure below.




We can describe all these scenarios as horizontally and vertically Data-Linked Functions and Devices. Here the computation (data analytics or simulation) and Machine learning are executing in tandem with the machine learning monitoring, tuning and learning results of the computation. The ML-computation interaction can be at the fine grain (function) component level. The breakup into functions leads to horizontal data flow in the figure within each of the two segments with vertical linkage to represent the ML-computation or the edge-fog-cloud linkage. As one builds more domain-specific hardware we can also see a vertical linkage between software and hardware units. Data flow is familiar for many successful workflow environments but the programming model just described requires fine-grain dataflow with greater performance challenges. Twister2 addresses this with Twister2:Net [3] high-performance communication subsystem that outperforms those in systems like Spark and Flink. It has four distinct message

systems: as well as dataflow DFW, it supports native MPI, a custom map-collective model Harp and the standard streaming publish-subscribe based approach. Each of these is suitable in different circumstances.

Twister2 assumes that it is integrated with separate high-performance simulation or data analytics such as that in Tensorflow. It is available for some demonstrations already as it offers the full Apache Storm or Heron streaming capability with built-in high performance for edge-fog-cloud use cases. It has two major additional capabilities that will be operational in 6 months; high-performance Python binding and connection to Apache Beam for the SQL query capability and orchestration offered by Beam. With these added, Twister2 will support a rich set of demonstrations and has a tutorial available [4].

The current Twister2 release supports resource provisioning in standalone mode, Kubernetes, Mesos, Slurm, and Nomad with several task schedulers and a web GUI for monitoring. The task graph module allows the creation of dataflow graphs for streaming and batch analysis including iterative computations with data caching. Twister2:TSet [5]supports distributed data with similar functionality to Spark RDD, Flink DataSet and Heron Streamlet. There are programming API's for streaming and batch applications with Apache Storm compatibility, as well as API's for operators, task graphs, TSets, and data and message level communication in different modes discussed above. Local file systems and distributed storages as in HDFS or shared NFS are supported while support for NoSQL and RDMS databases are planned for upcoming releases. Think of Twister2 as Apache Spark or Flink with high performance built in from scratch and currently able to support streaming computations, data operations in batch mode, dataflow and iterative computations.

The extensive messaging seen in the picture suggests the importance of hardware that supports high speed messaging and disk access. Natural hardware would be Omni-Path or Infiniband networking, Optane style memory and NVMe disks. The CPU's should support both machine learning and computation with both running close to each other and not separated in distinct clusters.

Questions Answered

1. *What innovative capabilities/functionalities will the proposed candidate platform demonstrate?*
   We are not aware of high-performance software environment offering the range of capabilities supported by Twister2
2. *What applications/communities would/could be addressed?*
   Initially we will focus on edge-fog-cloud applications as well as the "learning everywhere" MLforHPC applications
3. *What is the "platform vision," i.e. what kind of shared cyberinfrastructure (CI) for science would the further research/design/development of this platform lead to?*
   This is described above as the integration of diverse functions or microservices across

ML and computation. It should enable integration of multiple application models with good fine grain parallel performance

4. *How available/ready/complete is the set of software components to be used to build the demonstrator?*
   It is available for experimentation (0.2.0 release [1]) now with broad capabilities supported 6 months from now.
5. *As far as one can tell at this early date, to what extent can this be done with existing and/or otherwise available hardware/software/human resources?*
   We listed the preferred hardware above but standard HPC clusters can be used especially if they have local SSD or NVMe disk storage
6. *What is the potential international footprint of the demonstrator?*
   We don't see any problems with international use. All software is available from Github with an Apache 2 license.

**References**


[1] "Twister2 Release," *Twister2 high performance data analytics hosting environment that can work in both cloud and HPC environments*. [Online]. Available: https://github.com/DSC-SPIDAL/twister2/releases. [Accessed: 26-Oct-2018]

[2] Kamburugamuve, Supun and Govindarajan, Kannan and Wickramasinghe, Pulasthi and Abeykoon, Vibhatha and Fox, Geoffrey, "Twister2: Design of a Big Data Toolkit," *Concurr. Comput.*, vol. EXAMPI 2017 workshop at SC17 conference, 2019 [Online]. Available: http://dsc.soic.indiana.edu/publications/twister2_design_big_data_toolkit.pdf

[3] Supun Kamburugamuve, Pulasthi Wickramasinghe, Kannan Govindarajan, Ahmet Uyar, Gurhan Gunduz, Vibhatha Abeykoon, Geoffrey Fox, "Twister:Net - Communication Library for Big Data Processing in HPC and Cloud Environments," in *Proceedings of Cloud 2018 Conference*, San Francisco [Online]. Available: http://dsc.soic.indiana.edu/publications/twister_net.pdf

[4] G. Fox, "Twister2 Tutorial at BigDat2019 Cambridge UK January 8-11 2019." [Online]. Available: https://twister2.gitbook.io/twister2/tutorial. [Accessed: 30-Jan-2019]

[5] Pulasthi Wickramasinghe, Supun Kamburugamuve, Kannan Govindarajan, Vibhatha Abeykoon, Chathura Widanage, Niranda Perera, Ahmet Uyar , Gurhan Gunduz, Selahattin Akkas, Geoffrey Fox, "Twister2:TSet High-Performance Iterative Dataflow," Apr. 2019 [Online]. Available: http://dsc.soic.indiana.edu/publications/dataflow_twister2_new.pdf

## Proposal for a BDEC2 Platform Demonstrator from CERN and SKA

We would like to propose a proof-of-concept container platform and batch integration for workloads submission to access HPC testbed resources for Data Intensive Science applications such as HEP (LHC experiments) and radio-astronomy (SKA).
Representatives workloads, which in HEP concern the so-called “offline processing” i.e. the asynchronous data processing workflows in WLCG [1] - reconstruction and simulation - which happen after the primary raw data has been collected, would include (in order of complexity):

1. A standalone containerized benchmarking suite, that we propose to measure the performance of compute resources for our workflows; this benchmarking suite contains representative applications by each experiment for both High Throughput Computing (HTC) and High Performance Computing (HPC); it is light-weight, extensible and can be very easily modified to add other applications, including ML specific ones, and is reproducible, portable, with no need for network connectivity.

2. By leveraging the ‘OCI Image spec’, this demonstrator will show how optimised container images can be distributed to get the most performance out of the different target systems.

These workflows are available and ready to be used for the demonstrator. At a later stage we would like to extend the work to include more complex workflows, such as:

3. Fast simulation for LHC Monte Carlo detector simulation; no input data is needed. Generative models approaches are used, in particular with 3D conditional Generative Adversarial Networks (GAN); here significant speed-up for the training process can be achieved in an MPI based distributed parallel approach. On the SKA side approaches based on graph-based analytics using DASK would also be of particular interest.

4. Once workflows 1 and 2 are demonstrated, a complementary and fully IO intensive workload could be also exercised, e.g. via a full reconstruction workflow of an LHC experiment, which requires accessing large amounts of raw data, significant local IO and significant data export to a secondary analysis data format, typical of a data intensive science, like HEP and the distributed predict I/O requirements of the SKA[2].

In terms of “platform vision” and shared cyberinfrastructure (CI) we would like to demonstrate how to integrate HPC facilities with our existing world-wide distributed computing facilities, which in a way are representative of distributed computing models such as found in both the

SKA Regional Centres and SKA-SDP [3] approaches which may involve the exploitation of more agile, cloud-like platforms to accommodate a range of different execution environments. We have considerable experience in integrating our applications within external resources such as public and private clouds. The LHC experiments have also integrated “ad hoc” into several HPC centres, in particular in the USA and Europe.

We believe that this demonstrator would add significantly in terms of a harmonized and common approach to integrating HPC facilities. We can perform the integration of this demonstrator leveraging existing human resources and using our existing distributed computing software. We would be looking for partners at HPC sites, in particular to access their, as well as potentially share, test-bed resources, sufficient at this stage for a proof-of-concept.

[1]
http://cds.cern.ch/record/1695401?ln=en

[2]
http://ska-sdp.org/sites/default/files/attachments/distributed_predict_io_prototype_part_1_-_signed.pdf
[3]
http://ska-sdp.org/sites/default/files/attachments/ska-tel-sdp-0000151_01_sdp_p3alaskareport_part_1_-_signed_2.pdf

# Demonstrator proposal for BDEC2 Poznan

Information Technology Center, The University of Tokyo

1. What innovative capabilities/functionalities will the proposed candidate platform demonstrate (e.g. transcontinuum workflow, edge computing, data logistics, distributed reduction engine, etc.)?
   - Provide the platform which enables real-time data acquisition, data logistics, and realtime data analysis with on-demand configuration, expansion, and integration as well as computational science and engineering (CSE)
   - Important to create synergy between computational sciences and cyberphysical applications on such platforms
2. What applications/communities would/could be addressed?
   - Traditional computational science applications
   - Various kinds of applications with large-scale data processing and realtime (streaming) data analytics: Disaster prevention, Medical, Smartcity, Agriculture and fishery, and so on.
3. What is the "platform vision," i.e. what kind of shared cyberinfrastructure (CI) for science would the further research/design/development of this platform lead to?
   - Provide infrastructure as the fusion of BDEC (Big-Data and Extreme-Computing) supercomputer system for large-scale simulation, data-analysis, and machine learning with external part to acquire data via the Internet, and Data-Platform system that enables responsive applications with higher security by isolation and encryption for each project than the external part of BDEC system in early 2021
   - Create extra values by circulating data provided by other users via the Data-Platform
   - Access to various edge and IoT devices through secure mobile network with VPN by SINET academic network
4. How available/ready/complete is the set of software components to be used to build the demonstrator?
   - Mainly Integration of existing software, such as resource management, job scheduling, container, and so on.
   - In collaboration among vendors, potential platform users, and ourselves
5. As far as one can tell at this early date, to what extent can this be done with existing and/or otherwise available hardware/software/human resources?
   - Hardware for such a platform will not differ from existing hardware. Some kind of extensions for security could be required.
   - Regarding software, there is no suitable products especially for resource orchestrator, so we need huge amount of developments. Common data exchange format and APIs are also needed.
   - Coordinator for data curation is crucial for data utilization and circulation.
6. What is the potential international footprint of the demonstrator?

- No actual collaborations yet, but the large-scale data analysis with global observation network is suitable for this demonstrator.
- Astrophysics, such as blackhole detection
- Global climate prediction with data assimilation, such as typhoons and hurricanes
- …

## 1. Introduction

Satellites produce a wealth of data and information regarding the Earth sub-systems (land, atmosphere, oceans, solid Earth and biodiversity) and cross-cutting processes (climate change, sustainable development and security).
Space agencies cooperate together in order to optimize the usage of their data; this is done for example in the context of the CEOS (Committee on Earth Observation Satellites). CEOS is for example a forum of technical exchange on data services interoperability (discover, access, subset, visualization & process), Future Data Architecture (datacube, cloud, analysis ready data, exploitation platform…), …

Most of space data are open and free (with an exception for very high resolution imagery or for some countries or for cooperation with private entities).
The volume of space data is increasing exponentially with some programs like Copernicus, SWOT, NISAR or the last generation of weather forecast satellites (GOES, HIMAWARI, MTG). It represents dozens of PB and will reach several hundreds of PB in the next 3 three years (For example, for NASA alone, the growth rate of the archive will be around 50 PB from mid-2022.).

Given the explosion of data, it is now advisable not to download large volumes of data at home, but rather to move the processing where large data are hosted.

## 2. CNES organisation and services

CNES is the French space agency. It operates several satellites (JASON, CFOSAT, SARAL, Megha-Tropiques, Calipso, SMOS, IASI, PLEIADES, …) and develops new ones (MERLIN, MICROCARD, IASI-NG, CO3D, …).

Most of the missions are done in cooperation with other space agencies (ESA, EUMETSAT, ISRO, NASA, NOAA, …). In terms of processing, CNES is only responsible for the despatialization of data; that is to say up to a product level that does not require an expertise of the satellite or its instruments. In some cases, depending on the cooperation agreements, the treatments are carried out by our partners. Depending on the case, CNES may be responsible for the distribution of products and their long-term archiving or it may delegate these activities to its partners.
CNES is also very committed to promoting the use of its data and spatial data in general.

Some processing can be executed in the satellites themselves to reduce the amount of data to be transferred to Earth (Edge Computing); this is the case, for example, for the IASI instrument on METOP. But this is not a widespread practice now.

CNES also hosts a mirror of Copernicus data (French Copernicus Collaborative Ground-Segment) which represents about 10 PB (4PB on line and 14 PB capacity on tape).

When a data is processed by CNES, it is on its own computer center. A presentation (made in July 2019) of the CNES computer center can be found here. It allows numerical simulations (HPC), but the data processing is made in a specific HTC/HDA environment. For data reprocessing (very resource intensive), it is foreseen to use cloud-bursting solutions on commercial means.

The goal of CNES and other space agencies is to promote the use of data by the widest possible user communities. These categories of users can be classified macroscopically into two broad categories: research and the downstream sector.

For the downstream sector, the preferred solution to further exploit the data is commercial cloud computing; This for any type of treatments, including AI. For that, there are a large number of solutions:

- In Europe, 5 CDIAS (Copernicus Data and Information Access Services) have been initiated by the European Commission. They propose processing capabilities (cloud computing), services, and additional data. They rely on commercial clouds; namely CloudFerro, Orange, OVH and T-System.
- Several SME initiatives – for example Sinergise Sentinel Hub or Terradue Ellip.
- Amazon and Google propose a wide range of satellite data
    - All sentinel 1&2 (Copernicus) data are already hosted by Amazon. US agencies (NASA, USGS and NOAA) are moving their data to Amazon to promote its use and facilitate treatment with very large data by users (research & downstream).


*Figure: DIAS concept*

For the research sector in France, satellite data are exploited in five thematic Data & Services hubs:

- AERIS for the atmosphere thematic
- FOR@MATER for the solid Earth thematic
- ODATIS for the ocean thematic
- PNDB for the biodiversity thematic
- THEIA for the land-surface thematic

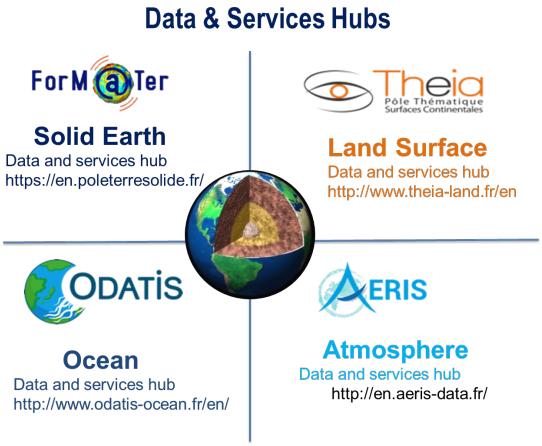



*Figure: Earth System Data & Services Hubs in France*

Each Data & Services Hub is geographically distributed across multiple data & services centers that all have their own computing capabilities (which mostly are HTC/HDA type clusters). They do not only deal with satellite data, but also a very large amount of in-situ heterogeneous data (ground, sea, airborne…). These data are less bulky than satellite data, but they are much more varied in terms of content, size and formats.

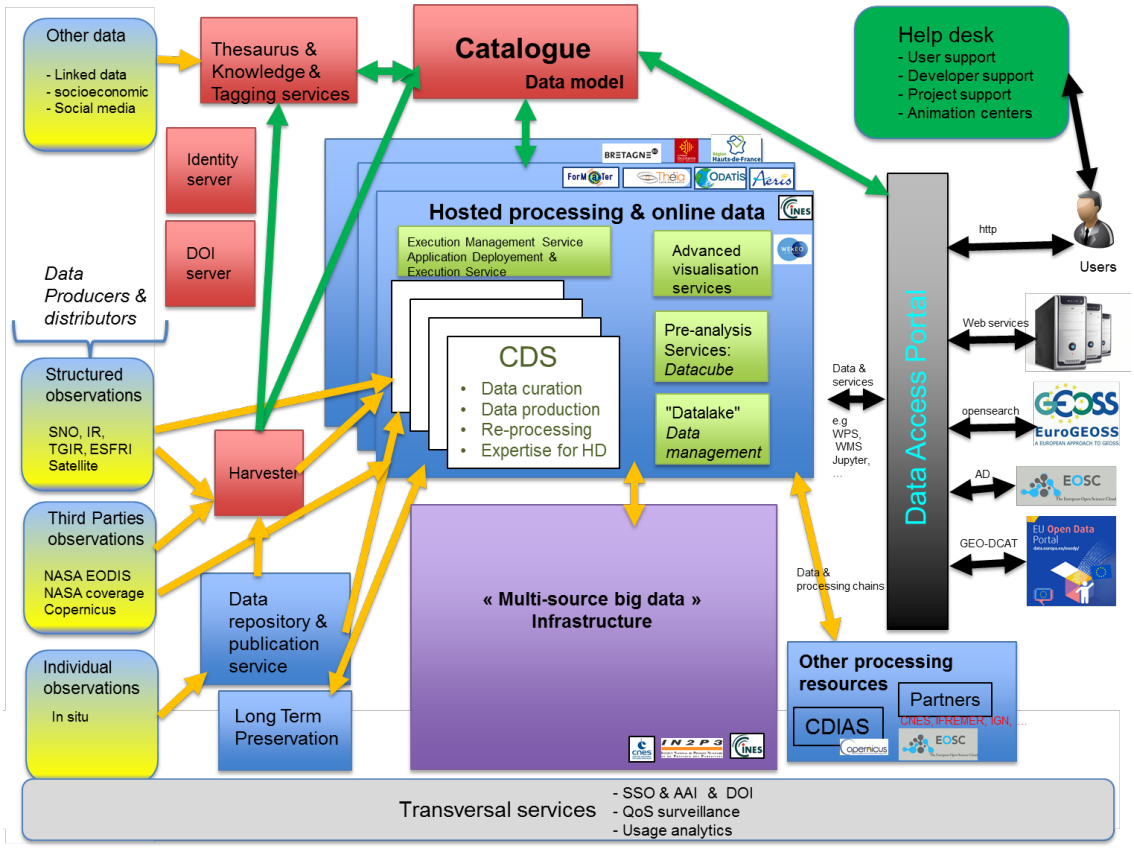
With the rise of IoT and 5G, the volume of in-situ data may explode. In this case it may be necessary to fully review the computing hierarchy of this data and turn to technologies such as Edge Computing, data compression/reduction, use of software defined networks for a smart orchestration of the successive level of resources, support of workflows from end to end (from the edge to the tape of the data center), …
In that sense, some Data & Services Centers use data from models that are processed in HPC centers located in other research infrastructures.

**Globally, our capacity to adopt an integrated inter- and trans-disciplinary approach is hampered by the fact that Earth Science data is today highly compartmentalised between these different scientific disciplines and communities.**

## 3. DATA TERRA : toward a fully integrated earth sciences data distributed platform

The Earth is a living system encompassing multi-scale and multi-physics internal dynamical processes and interactions with its external fluid envelopes (e.g., ocean, atmosphere) and continental interfaces (lands, biosphere and anthroposphere). Understanding, monitoring, and predicting the evolution of the Earth's systems in their environments is a fundamental scientific challenge with important societal and economical applications in terms of natural hazards (e.g. volcanoes, earthquakes, tsunamis, landslides), environmental and climate change, new energy resources, sustainable development.

Some countries have undertaken major efforts to restructure this heterogeneous ecosystem, mutualise resources and expertise, and to provide platform of services enabling from end-to-end (edge to the tape) the efficient data logistics including discovery, access, interoperability and wider reuse of data within and beyond Earth Science communities, to society as a whole. The French “DATA TERRA” Research Infrastructure is an example, and a recognized global leader. System Earth was established in France in 2017 to integrate existing data and service hubs and provide easy access to Earth Science data and associated products across the board for scientists and decision makers. With the core mission to facilitate and foster integrated & interdisciplinary research to understand DATA TERRA processes and Global Changes, DATA TERRA is committed to implementing the FAIR principles, creating distributed services, tools and workflows for data management, curation and scientific use; and promoting dialogue and international agreement on best practice.




*DATA TERRA High level Architecture*

DATA TERRA will have to face several challenges:

- the level of FAIRness of the different centers is heterogeneous
- Bulky data are geographically distributed in heterogeneous computing infrastructure with different level of services; there is a need to develop crosscutting applications then to combine different data in different location (in a distributed software platform) with implication on:
    - Workflows
    - Data logistics
    - Network
    - Computing infrastructure
- Stay open to European cooperation (e.g. ESFRI & ENVRI) and international cooperation (e.g. RDA)
- It will not be possible to build a monolithic and centralized system with all data and processing resources
- In a context of convergence of HPC, HPDA and AI, take into account the moving computing infrastructure landscape in France and in Europe
    - Existing processing capabilities in the Thematic Data and Services Hubs
    - French INFRANUM project led by the French Ministry of Research to concentrate the processing infrastructures at the regional and at the national level (GENCI)
    - EOSC which is the natural solution for ESFRI
    - DIAS

PRACE the HPC European Research Infrastructure, EuroHPC and EDI (European Data Infrastructure)

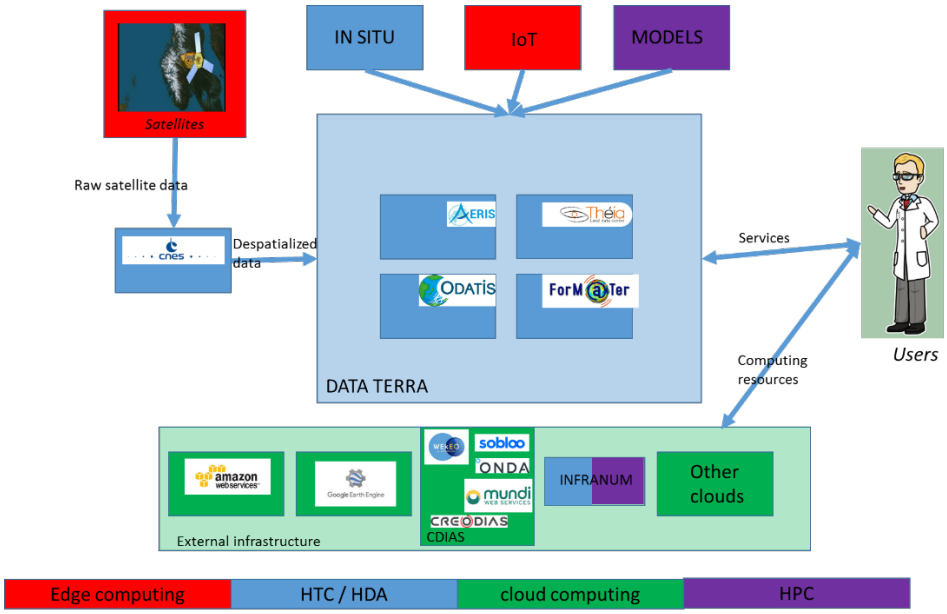



Figure: TERRA DATA transcontinuum workflow

**The proposed BDEC proposal is a prototype of an architecture that will allow DATA TERRA to fulfill its objectives.**

1. What innovative capabilities/functionalities will the proposed candidate platform demonstrate (e.g. transcontinuum workflow, edge computing, data logistics, distributed reduction engine, etc.)?
The prototypes addresses transcontinuum workflows (cf figure above), edge computing, data logistics.

2. What applications/communities would/could be addressed?
The communities are the Earth Science communities.

3. What is the "platform vision," i.e. what kind of shared cyberinfrastructure (CI) for science would the further research/design/development of this platform lead to?
Cf two previous figures/

4. How available/ready/complete is the set of software components to be used to build the demonstrator?
To be developed. The CEF (Connecting European Facilities) OpenData/HPC EC project PHIDIAS (duration 3 years from July 2019 with 16 European partners including CNES, CINES, CSC, CERFACS, IS Terre, IRD, IPSL, SPACIA, …) will allow to develop the first elements of the system.

5. As far as one can tell at this early date, to what extent can this be done with existing and/or otherwise available hardware/software/human resources?
Cf point .4

6. What is the potential international footprint of the demonstrator?
System Earth is by nature an international topic. It can be derived at European level in the frame of ESFRI/ENVRI. It can be derived at international level in the frame of GEO or RDA.

# PNSC demonstrator

A brief description of this platform, according to your questions, is provided below. It would be great if we could show it during the meeting.
Thank you in advance for your feedback on this.
If there is also any way in which we could support this BDEC initiative please let us know.

1. What innovative capabilities/functionalities will the proposed candidate platform demonstrate (e.g. transcontinuum workflow, edge computing, data logistics, distributed reduction engine, etc.)?

The Innovation Driven Computing platform provides an architecture and set of tools enabling execution of applications in containerised environments both at HPC center and edge level. It includes own solutions for:
- Distributed workflow management & application execution and coupling (new version of QCG software stack)
- Advanced visualization and in-situ steering (QCG Now and VR tools),
- Extreme scale and energy efficient application modelling (DCworms)
- Lightweight data and digital objects management (Ibis)
- Research Objects and metadata management platform

Computations can be run at large centralised systems as well as executed efficiently at edge nodes. Our recent research achievements in the area of microservers and edge computing deliver a set of dedicated and optimized systems such as:
- Powerful AI processing
- Autonomous and energy efficient processing
- Automated and adaptable power capping
- Thermal management

2. What applications/communities would/could be addressed?

The applications that are addressed by the platform include:
- Applications based on numerical weather forecasts, e.g. prediction of air quality in cities or renewable energy production
- AI for industry4.0: analytics and predictive maintenance in electrical vehicles and smart factories
- Analytics and services for energy sector and smart grids
- Other industrial applications, e.g. design (of furnitures)
- Agriculture, e.g. detection and prevention of diseases

The aim of the platform is to address (in addition to traditional scientific applications) key industrial sectors that may take advantage of convergence of edge, cloud and HPC technologies.

3. What is the “platform vision,” i.e. what kind of shared cyberinfrastructure (CI) for science would the further research/design/development of this platform lead to?

The platform is built around a vision of Innovation Driven Computing. It addresses the computing continuum from edge to clouds/HPC by optimised distribution of data and computing across these layers. To achieve this goal it combines tools for execution of computing and data processing in HPC systems, Platform as a Service (PaaS) interfaces, and machine learning and data analytics tools.
It assumes the use of existing computing, storage and network infrastructure built for large scientific applications and enhance it with cloud functions that facilitate development of industrial applications and with integration of edge systems.

4. How available/ready/complete is the set of software components to be used to build the demonstrator?

The platform leverages on existing tools and applications developed over last decade by PSNC within EU research projects and industrial cooperations, complemented by the relevant open source projects. These software components are mainly based on the QCG software, PaaS system and microserver managment developed within M2DC project (http://m2dc.eu), and other tools listed also in point 1.

5. As far as one can tell at this early date, to what extent can this be done with existing and/or otherwise available hardware/software/human resources?

The demonstrator can be done integrating HPC data centers, in particular PSNC supercomputing and edge resources/testbeds, and a consistent PaaS core platform with various new capabilities. Testbeds include a micro data centre laboratory where new systems can be deployed and tested, and remote labs such as FutureLab Living Laboratory or the (under construction) “Airport of Things” lab that might allow tests of edge systems for drones, robots, and other unmanned vehicles.

6. What is the potential international footprint of the demonstrator?

The demonstrator will use tools and applications developed or used in multiple international consortia with large multinational companies, renown universities and research centres etc. Among others, part of the demonstrator will be developed and used in the national PRACE-LAB project that aims at large distributed HPC and data processing platform. The demonstrator will reach communities of EU projects such as M2DC or FET-HPC ones: VECMA, ASPIDE, RECIPE. The significant participants of these initiatives include companies such as ARM, Bull, Vodafone, Huawei, as well as biomedical, numerical weather forecast communities and many others.

# BDEC2 Platform Demonstrator Proposal

## 1 Overview

The BDEC2 effort has identified the need for a demonstrator of a potential platform for programming the computing continuum that is the future of big data and extreme scale computing. A recent paper by BDEC2 organizers and participants[1] describes the notion of the continuum, enumerating classes of computing elements from *nano* (count = $10^9$, size = $10^1$) to *facility* (count = $10^1$, size = $10^9$).

This proposed contribution to the platform demonstrator is based on the InLocus architecture[2], which operates on streams of data. Due to its limited execution model and semantics, it is suitable for processing streams of e.g. sensor data on a message by message basis. This primitive execution allows InLocus to execute on devices at the *nano* or *micro* end of the computing continuum.

By processing data a message, or tuple, at a time, it resembles cloud data processing systems like Apache Storm or Twitter Heron, both of which assemble topologies of stream processing functions ("bolts") and route messages via these processing pipelines. These processing elements can be considered analogous to "microservices" in a serverless computing environment. This type of decomposed application design pattern underlies web-scale applications running at the Cloud end of the computing continuum.

Finally, there is a growing movement in high-performance computing to move to Asynchronous Many-Task (AMT) runtimes to expose parallelism. Applications can be defined in terms of a "dataflow" model, where small

[1] P. Beckman, J. Dongarra, N. Ferrier, G. Fox, T. Moore, D. Reed, and M. Beck. Harnessing the Computing Continuum for Programming Our World, `https://www.researchgate.net/publication/332246123_Harnessing_the_Computing_Continuum_for_Programming_Our_World`

[2] L. Brasilino, A. Shroyer, N. Marri, S. Agrawal, C. Pilachowski, E. Kissel, and M. Swany. Data Distillation at the Network's Edge: Exposing Programmable Logic with InLocus. In IEEE International Conference on Edge Computing, July 2018. `https://doi.org/10.1109/EDGE.2018.00011`

"patches" of computation are performed when the requisite data is available. This needs to be done carefully as over-decomposed structures may not efficient, but the model is the same as that above – small computational operators await an input of some sort and then execute over it.

## 2 Proposed Platform Demonstrator

Each of the three above three environments achieves scalability by decomposing applications into ensembles of small operations, working together. Given that this data-driven, message-oriented model can be productively implemented across the computing spectrum, we assert that it can enable a *transcontinuum workflow* with explicit *edge computing* support.

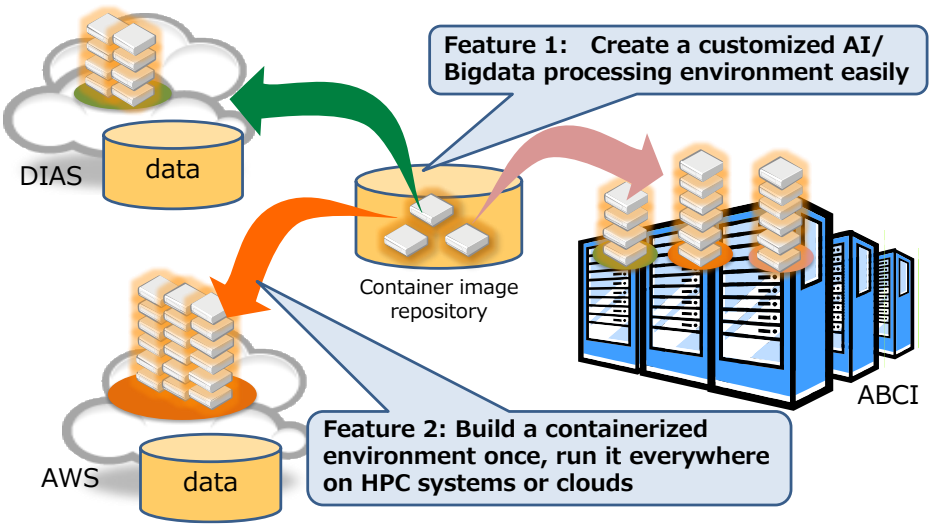
A straightforward application would be processing sensor data. This would be feasible to implement using combinations of various BDEC2 stakeholder's projects. An environmental sensor data application running across various scales of the computing continuum is a powerful vision for the convergence we seek. Further, early demonstrations of this are possible by combining existing components.

Proposal to a BDEC2 Platform Demonstrator

Contact Person: Ryousei Takano <takaon-ryousei@aist.go.jp>
National Institute of Advanced Industrial Science and Technology, Japan

**[Overview]**

We propose a demonstration of distributed and containerized AI/Bigdata processing environment in "build once, run everywhere" manner. We have constructed and operates ABCI, an open innovation platform for advancing AI research and deployment [1]. ABCI allows the user to submit a large-scale distributed deep learning job on compute nodes through Singularity container engine and Univa Grid Engine [2]. We expand it on a global scale for research reproducibility and productively. Our primary application is satellite image processing using deep learning techniques (e.g., [3]). Each institution has huge amount of image data and container images are shared with them for extracting knowledges and insights from such bigdata. We just started collaboration with international partners in this field. It is a very early stage of software development.




**[Answer for questionary]**

1. What innovative capabilities/functionalities will the proposed candidate platform demonstrate (e.g. transcontinuum workflow, edge computing, data logistics, distributed reduction engine, etc.)?
   - Singularity container on HPC systems
   - A data access abstraction layer to seamlessly access data from a container to underlying storage systems such as parallel file system and object store.
   - A persistent identifier (PID) management system for data, workflows, and container images. It is important to guarantee research reproducibility.
2. What applications/communities would/could be addressed?

- Geoscience and remote sensing. But this idea is not limited for this area.

3. What is the "platform vision," i.e. what kind of shared cyberinfrastructure (CI) for science would the further research/design/development of this platform lead to?
   - A distributed and containerized AI/Bigdata processing environment in "build once, run everywhere" manner
4. How available/ready/complete is the set of software components to be used to build the demonstrator?
   - The software development is undergoing.
5. As far as one can tell at this early date, to what extent can this be done with existing and/or otherwise available hardware/software/human resources?
   - Uncertain.
6. What is the potential international footprint of the demonstrator?
   - UCSD/Pacific Research Platform
   - NCHC, Taiwan

**[References]**

[1] ABCI, https://abci.ai/ (2019)

[2] H. Mikami, et al. "Massively Distributed SGD: ImageNet/ResNet-50 Training in a Flash," https://arxiv.org/abs/1811.05233 (2019)

[3] K. Enomoto, et al. "Image Translation Between Sar and Optical Imagery with Generative Adversarial Nets," IGARSS 2018 (2018)

# Data Processing in Radio Astronomy - Preparing for the Square Kilometre Array

## White Paper for BDEC2 meeting in Poznan – May 2019

Prepared by:
M.P. van Haarlem (ASTRON), J. van Leeuwen (ASTRON/Amsterdam), J.B.R. Oonk (SURFsara/Leiden/ASTRON), T. Shimwell (ASTRON/Leiden), L.V.E. Koopmans (Groningen)

This document summarizes three use cases of computationally intensive LOFAR[1] data processing and analysis workflows. These demonstrate the kind of processing required for the Square Kilometre Array (SKA) radio telescope, which is expected to be fully operational from 2026/2028. The workflows presented all require further development and new services in order to deal with the increased data volumes expected in the operational phase of the SKA. These developments strongly challenge the convergence between HPC, HTC and HDA data and computing capabilities, and their integration in the context of a software platform of distributed services across a continuum of edge and centralized (Cloud, HPC) infrastructures to support such complex, wide-area workflows and their data logistics.
In the operational phase, SKA will involve multi-stage, and geographically distributed science-driven data processing and analysis workflows. The first stage in the two telescope host countries (Australia and South Africa) will deal with the processing and reduction of raw multi-source data streams across edge (close to the instruments) and centralized (HPC) infrastructures to deliver first stage (Observatory) multi-type Data Products – i.e., event and visibility data. These primary data products will be distributed to a network of ~10 geographically distributed SKA Regional Science Centres[2] that archive the data and where the second stage of the science application high-end data processing/analysis takes place. The total data volume transported to and archived in these regional centres will grow to a rate of ~1 Exabyte per year from ~2027/8 and raise challenging and potentially disruptive issues in terms of data logistics, i.e. the management of the time sensitive positioning and encoding/layout of data, relative to its intended users and the computational and data resources and services that they can apply. In many countries the SKA Regional Science Centres will make use of common data and computing infrastructure shared with other data/compute intensive disciplines.

The H2020 ESCAPE[3] is a first step towards the collaboration between partners from astronomy and particle physics in the context of the European Open Science Cloud (EOSC). ESCAPE aims at delivering solutions to ensure integration of data, tools, services and scientific software; to foster common approaches to implement open and FAIR data services and stewardship; to establish within EOSC interoperability services enabling an integrated multi-probe facility for fundamental science. Integration of a continuum of existing European edge and centralized (HPC, Cloud) infrastructures into a software platform of distributed services will foster the convergence between HPC, HAD and HTC (including data-driven Machine Learning and AI), and shall provide an enormous boost to the discovery potential through the interoperability of planned large science-driven infrastructures. As such this can contribute to a science-driven demonstrator for BDEC2 shared not only between astronomy (SKA) and particle physics (CERN/LHC) communities but also other communities such as Space Observation.

Disruptive issues in the coming decade for LOFAR/SKA:

1) Any large scale (KSP) type projects, with a semi-continuous data flow, must move to a workflow solution on compute near the data. This to some degree means professionalizing the current codes, embracing CI/CD developments to orchestrate update and community input, and it means giving up (some of) the freedom of doing data reduction on your own laptop/mini-cluster.

[1] https://www.astron.nl/telescopes/lofar
[2] Design activities for these regional centres are ongoing in most of the SKA member countries. In Europe these are being coordinated in the H2020 AENEAS project (www.aeneas2020.eu).
[3] ESCAPE brings together the following ESFRI facilities (CTA, ELT, EST, FAIR, HL-LHC, KM3NeT, SKA) as well as other pan-European research infrastructures (CERN, ESO, JIV-ERIC, EGO-Virgo) in the astronomy and particle physics research domains.

2) Very high memory imaging. We will breach the 20x20k image size in the next decade and move towards 100kx100k images. With current DD(F) pipelines this will require about 2 TB of memory on a single node to run efficiently. New types of parallelisation which can spread the DD(F) calibration/imaging load across multiple nodes may lower this requirement.

3) High levels of automated quality control to lower the pipeline failure rate - however this may mainly affect the initial calibration stages. ML-assisted auto-tuning may play a role here, but it still needs to be developed. To what extend these methods may also play a role in the later DD calibration/imaging is not yet clear (i.e. can optimised cal/img strategies be found based on (i) data and (ii) astronomers reqs (e.g. diffuse emission and combining it with high resolution emission - how to (simultaneously) image scales that span more more than 2 orders in magnitude in baseline length?)

4) On-demand resources in terms of fat nodes for radio astronomy are very expensive. Tuning the workload and the defining interaction models between astronomers and the underlying hardware needs to be investigated (cloud vs. batch, or both?)

5) Grid and X509 solutions must be updated to cloud (probably managed cloud service layers and not self-service) and AAI solutions in the coming years - this requires coordination across the edge to continuum of resource providers, observatories and user(institutes).
Data storage models will also change to data lake-type models. This will require a new layer of organisation or alternatively abstraction. Similarly workflows will need to run in a seamless manner across infrastructures, which implies that they either need to be independent of underlying or be made aware via intelligent middleware layers.

6) The radio astronomy case, in particular SKA, will break open a new realm in compute, that is fundamentally different from HEP in terms of data organisation and processing requirements. Whereas HEP workflows consist of many small event files that can be treated independently, the opposite is true for the SKA in that a through complex set of calibration tasks all data needs to converge at several instances in this workflow, i.e. the data is not independent (although some parts of it can be treated via diverging flows before ultimately converging again).

## Case 1: Pulsars

Pulsars are fast spinning, highly magnetized neutron stars that provide physical conditions far beyond those attainable in Earth-bound laboratories. Studying their extremes is thus key to understanding fundamental gravitational and particle physics. The SKA will capitalise on digital signal processing and computing power to provide an unparalleled field-of-view. The SKA will survey the sky for their periodic, single-pulses (Fast Radio Bursts) and slow-transient emission.

Finding radio pulsars in radio-telescope data is exceedingly data and computationally intensive. Data streams from the SKA stations and dishes are first pre-processed on dedicated edge FPGA hardware. The resulting streams are combined in beamformers on site and in centralised facilities in the host countries. Data streams are split into independent pointings that need to be further processed and analysed individually outside the host countries in a network of regional centres.

We will be performing massive surveys with SKA; for both their periodic, single-pulse (Fast Radio Burst) and slow-transient emission. These searches are a massive data and computational challenge across a continuum of edge and centralised (HPC, Cloud) infrastructures raising disruptive issues in term of distributed data and computing services, as well as in term of data logistics across this continuum.

Pulsar Processing Details

**1) Data preparation – creation of filterbank files**

This is a highly disk I/O–intensive process, where many >10-GB size files are read from a scratch disk pool or from the incoming network, re-sampled, and written back to shared storage. This requires a central data handling location on the input side of the HPC/Cloud centralised facility.

**2) Data preparation – Radio frequency interference (RFI) excision**

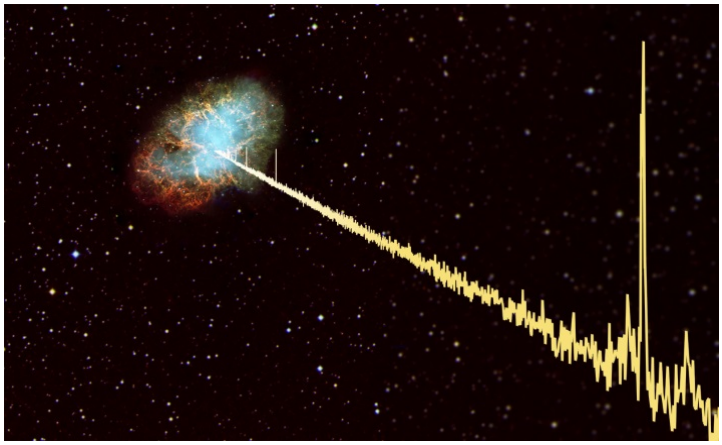
In interference excision, samples in a 2D matrix that strongly deviate from the mean are identified and removed. This is a more computationally expensive process, where all data is read and analysed in order to create an RFI mask. The (current) best performance is achieved by embarrassingly parallel farming over nodes.

**3) Search for periodic signals and single pulses**

The main algorithms for this stage are:

a. Dedispersion (progressively shifting all rows in a 10 GB 2D matrix, and collapsing to 1D)
b. Discrete Fast Fourier Transforms (FFTs)
c. Matched filtering – both in the time and frequency domain
d. Folding (chopping the data into similarly sized chunks and adding, algorithmic sped up using FFAs)
e. Machine learning algorithm for candidate scoring and classification

For pulsar and FRB surveys with the LOFAR (Stappers et al. 2011; Sanidas et al. 2019) and Apertif (Maan & van Leeuwen 2017) telescopes, pathfinder instruments for the SKA, the processing in steps 1-3 amounts to 1 and 5 PFLOPS respectively. In each, about 1 PB of data is recorded and processed per day. These are powered by the Dutch national supercomputer Cartesius, and by a dedicated GPU/CPU cluster. For SKA, the planned compute required is of order 100 PFLOPS.



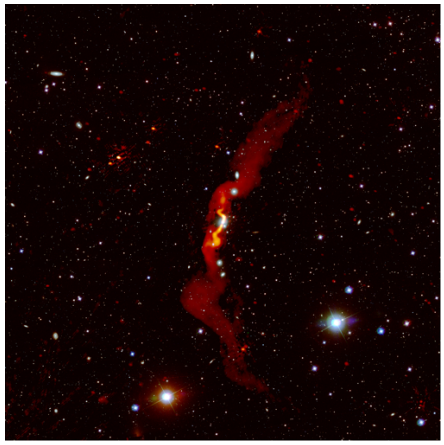


**Figure 1 (left):** A 1-second snapshot of the radio peaks that the Crab pulsar emits. In the background the outer layers that were ejected in the supernova are visible in the optical (ASTRON Westerbork/ESO VLT composite image).

**Figure 2 (right):** The radio galaxy 3C31, observed with LOFAR by Heesen et al (2018), is shown in red on top of an optical image. LOFAR reveals the radio galaxy to be more than 3 million light years in size. Credit: Volker Heesen and the LOFAR surveys team.

**References**

Coenen et al. 2014, A&A, 570, 60
Maan & van Leeuwen, 2017, Proc. URSI GASS 2017
Stappers et al. 2011, A&A, 530, A80

## Case 2: LOFAR Surveys Imaging Workflow

The Low Frequency Array (LOFAR) is a technology pathfinder for SKA-Low. The pipelines, developed by the Surveys KSP (SKSP), produce thermal noise limited images at 6 arcsec resolution (see Shimwell et al. 2019) and are able to process data in a semi-continuous flow (on average 1 new dataset per day). The challenges in imaging and calibrating LOFAR data are similar to those that will be faced by SKA-low and as such the SKSP imaging pipelines provide a very good demonstrator for processing future SKA-Low continuum datasets. Furthermore, these pipelines have already proved capable of processing deep LOFAR observations with over 100hrs of LOFAR data.

The LOFAR surveys project will observe approximately 3,100 fields (1,300 already complete). Each field must be processed as described below.

**1) Archiving the observations**: LOFAR surveys data are recorded with a resolution of 64ch per 195 kHz subband and 1s integration times. This results in approximately 64TB of data per pointing and two pointings are observed simultaneously in an 8hr period and hence collecting ∼128TB of data in 8 hrs. These data are flagged for interference and averaged to a resolution of 16ch per 195 kHz subband and 1s integration time and then ingested into the LOFAR archive. The resulting archived data has a size of 16 TB per pointing. This initial processing is completed on the CEP4 compute nodes and to date 450, 50, 800 datasets have been stored in a federated data archive that is hosted by SURFsara (Amsterdam), PSNC (Poznan) and FZ-Jülich respectively. In total the archived survey data will occupy approximately 30PB.

**2) Correction of Direction-Independent (DI) Effects.**: The archived LOFAR data must be calibrated to remove time independent (or slowly varying with time) instrumental effects such as the amplitude corrections required to get the flux scale correct, the offset between XX and YY phase and the clock offsets for each station. The procedure followed for this is described in van Weeren, et al. (2016), Williams, et al. (2016), Shimwell, et al. (2017) and de Gasperin, et al. (2019). The output data from this pipeline for each pointing are kept at a resolution of 2ch per 195 kHz subband and 8s integration time and occupy approximately 250GB and are stored on the SURFsara facilities. This processing is completed on the compute nodes at either SURFsara or FZ-Juelich (see Mechev et al. 2017) where the majority of the data are stored (data from Poznan are copied to SARA) and requires approximately 1000 CPU core hours per pointing. This step cannot be performed elsewhere due to limitations in transporting the large amounts of data.

**3) Correction of Direction Dependent (DD) Effects**: Low frequency radio data are severely corrupted by time and position dependent ionospheric errors which must be corrected in order to produce science quality images. Correcting these corruptions is still very much an area of active research. The surveys project uses a pipeline that makes use of *kMS* (Tasse 2014b) to calibrate for ionospheric effects and errors in the beam model and *DDFacet* (Tasse et al. 2017) to apply the derived solutions during the imaging. This pipeline takes about 5000 CPU core hours (5-7 on a single HTC node) days to image one survey pointing when operating on a compute node with 192 GB RAM (the minimum required for the pipeline is 192 GB) and two Intel Xeon Gold 6130 CPU that have 16 cores each and run at 2.1 GHz. Each observation requires 2-3TB of storage for input data and intermediate products. The surveys project makes continuous use of 10-20 of such nodes spread between Hertfordshire (the LOFAR-UK cluster), Leiden, Bologna, Hamburg and SURFsara. Once the pipeline is complete, the data, final calibration solutions and images are uploaded to a storage cluster in Leiden. These final DD legacy products for each pointing occupy approximately 250GB and the total volume required to store the entire survey will be approximately 800TB. The pipeline used for this step is described in Shimwell et al. (2019) and Tasse et al. in prep.

**Technical processing solution**
The LOFAR archival data uses distributed storage that the SKSP accesses via X.509 certificate-based federated AAI. The compute clusters at SURFsara and PSNC are also accessed via these certificates whereas the other compute cluster require individual ssh key-based solutions.

The SKSP workflow is based on containerized (*Singularity*) software that is built using continuous integration via *SingularityHub* and *Github* and deployed via the *Softdrive* virtual drive. Workflow orchestration and monitoring is done via the *AGLOW* package, consisting of the *Grid_LRT* framework (Mechev et al. 2017a) and

an Airflow-based workflow engine (Mechev et al. 2018b). To optimize the workflow the *Pipeline collector* profiling package (Mechev et al. 2018a) runs concurrently with the pipeline at less than 1% of the total CPU cost. To date ***7 Petabyte*** of surveys data have been processed using this solution and made available via web-based repositories at SURFsara and Hertfortshire.

**Future perspectives & SKA1-Low**: SKA1-Low will provide 3 times the bandwidth of LOFAR and about 9 times the number of stations. Simple scaling implies that size of the data products will increase by a factor 200 relative to the numbers mentioned above. i.e. the DI and DD products will each move from 250GB to 50TB. Note that DYSCO compression will be able to reduce this number by a factor 3-4 and hence whilst these products are large it is not impossible to envisage moving them between system with sufficiently fast data connections.

The automation and workflow orchestration of the current LOFAR Surveys processing pipelines rely massively on the underlying Grid infrastructure and associated tooling. This Grid model will need to change in the coming years to accommodate the requirements of the high-luminosity LHC and the SKA. These changes are a topic of vigorous discussion and we are now faced with the daunting task of shaping the new environment for these instruments. However, the need for this new eco-system also provides an opportunity for the astronomical community to re-think their solutions and make use of the latest technologies to, (i) integrate compute and storage models from the edge to centralized HPC/Cloud facilities and data lakes, (ii) harvest and deploy software using continuous integration and continuous delivery standards in collaboration with the community. This will undoubtedly change the way in which (radio)astronomy is carried out in the next decade.

In the SKA-era not only the data, but also the software and the users must move to the centralized compute facilities and these facilities must be able to cope with the diverse needs of the radio-astronomical community. The data sizes, organization and complexity of radio astronomy observations with LOFAR and the future SKA (100x LOFAR) also break open a new realm in compute and workflow requirements. Whereas high energy physics (HEP) case consists of a limited set of instruments with many small event files, the SKA case will provide a large set of observing modes and thus a very diverse output in terms data types, sets and sizes.

The SKA imaging case, as prototyped by the LOFAR Surveys, shows that the radio astronomy data cannot be treated fully independently and consists of complex workflows with diverging and converging data flows. Orchestration and (auto)-tuning, possibly assisted by machine learning, of these workflows across different infrastructures (geographically, platform & hardware) poses one of the greatest challenges for the SKA. In addition, further optimization of the underlying algorithms to become fit for purpose in 2025 is another challenge. As an example, we will here mention the imaging task of the pipeline that in the coming years will moving from 20k x 20k pixels to 100k x 100k pixels. If this task is performed will current algorithms the RAM memory requirements will increase from 192 Gigabyte to more than 2 Terabytes of memory.

**Breakdown of resources used across the three stages listed above**:

- CPU corehours (unit: hrs) -> 1:2:3 = 300:1000:5000
- RAM peak memory (unit: GB) -> 1:2:3 = 16:64:192
- Scratch storage (unit GB) -> 1:2:3 = 400:80:3000

This clearly shows that part-3 the most intensive. Here the CPU corehours correspond to the total of all parallel jobs belonging to one of the three stages, whereas as the RAM and scratch are the amount per job (for the dominant task). It is important to note that the parallisation of the parts is different. For SKSP processing, stage-1 consists of 488 jobs, stage-2 of 982 jobs and stage-3 of a single job. A breakdown of the processing in part-3 is shown in Figure 3.

The pie chart below shows each of these different steps and the time taken for **Part 3**, the direction dependent calibration and imaging stage. The processing starts at the top of the pie chart and to complete the entire process takes 5-7 days (the hatched region shows 1 day). The white regions are imaging and the dark regions are calibration with kMS and DDFacet respectively. The black regions at the end are the creation of auxiliary products (things like QU cubes and low resolution images) once the calibration is complete.

The inner bit of the pie chart gives the total breakdown of imaging (white), calibration (grey) and auxiliary (black).

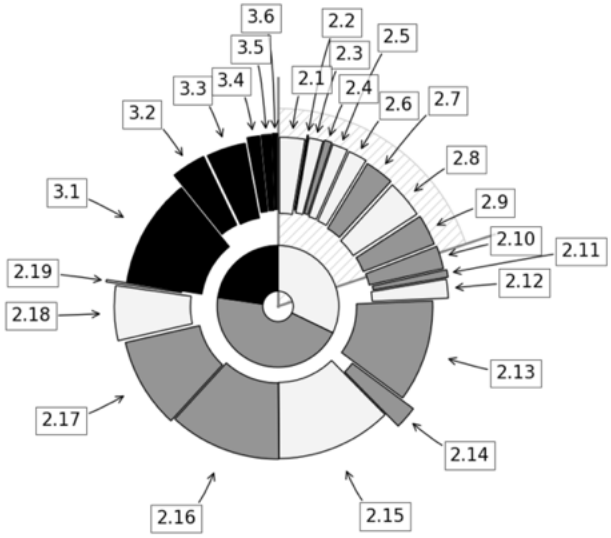



**Figure 3:** The pie chart below shows the different steps and the time taken for **Part 3** of the surveys workflow, the direction dependent calibration and imaging stage. The processing starts at the top of the pie chart and to complete the entire process takes ~5-7 days (the hatched region shows 1 day). The white regions are imaging and the grey regions are calibration with kMS and DDFacet respectively. The black regions at the end are the creation of auxiliary products (e.g. QU cubes, low resolution images and dynamic spectra) once the calibration is complete. The inner part of the pie chart gives the total breakdown of imaging (white), calibration (grey) and auxiliary (black).

**References**

de Gasperin F., et al., 2019, A&A, 622, A5
Heesen V., 2018, MNRAS, 474, 5049
Mechev A., et al., 2017, Proceedings for International Symposium on Grids & Clouds, 2
Mechev A., et al., 2018a, A&C, 24, 117
Mechev A., et al. 2018b, arXiv, arXiv 1808.10735 (IEEE)
Offringa A. R., 2016, A&A, 595, A99
Shimwell T. W., et al., 2017, A&A, 598, A104
Shimwell T. W., et al., 2019, A&A, 622, A1
Tasse C., 2014, arXiv, arXiv:1410.8706
Tasse C., et al., 2018, A&A, 611, 87
Tasse C., et al., in prep
Williams W. L., et al., 2016, MNRAS, 460, 2385
van Weeren R. J., et al., 2016, ApJS, 223, 2

## Case 3: Epoch of Reionisation

LOFAR, one of SKA's official pathfinders, has a similar station size, field of view, frequency and baseline-coverage as SKA-Low. The LOFAR-EoR Key Science Project (PI: Koopmans) data-processing pipeline (Figure 4) operates on a significant (132xNVIDIA-K40) GPU cluster and is partly distributed, operating at nearly 100% capacity. *Hence we deem the operational LOFAR-EoR processing pipeline as an excellent , but still limited, starting point for an SKA-Low simulation, processing and analysis pipeline*.

Below is a brief description of the modules implemented in the current LOFAR EoR processing pipeline shown in Figure 4.

**Direction Dependent Calibration (Stage 4) take >95% of all resources for LOFAR, so is by far the most intense**. For example imaging can be done in a few minutes, but calibration takes days. For SKA-low Stage 4 might take 99%+ of all resources. In order to have a major impact the focus should be on fully distributed near real-time calibration (e.g. on GPU-like architectures).

**1) Radio-Frequency Interference (RFI):** The first module in data processing is to excise ('flag') bad data, e.g., from man-made RFI. We use the AOFlagger module, based on signal-processing and morphological algorithms, to excise input RFI signals from the simulated SKA-Low data.

**2) Data Averaging**: Very high time/frequency resolution data, from the correlator, is averaged to lower resolution to obtain a manageable data volume, using the NDPPP module

**3) Direction-Independent (DI) Effects:** Instrumental (amplifier, cable-reflection) errors, after combining signals from receivers in a station ('beam forming'), and large scale ionospheric errors, are direction independent, meaning that each source in the sky is affected by it in the same manner.

**4) Direction-Dependent (DD) Effects:** Direction-dependent errors are different for sources seen in different directions on the sky. They can be due to amplifier gain errors in the receivers *before* beam-forming (see step 3), malfunctioning receivers, and the ionosphere.

**5) Wide-Field Imaging:** Low-frequency imaging is 'all-sky' imaging. We use the modules WSClean on CPUs and ExCon on GPUs, developed by Offringa and Yatawatta, respectively, to generate extremely wide-field data cubes (>30x30 degrees or $>10^9$ pixels per channel) over the full frequency range. These images form the input of the sky model building module (Step 6).

**6) Sky Model Building:** Both DI and DD calibration of the instrument and ionosphere (steps 3 and 4) require a precise and accurate 'sky model' (i.e. a set of parameterized source models with their frequency dependence). This sky model is built from the image cubes generated in step 5. The current sky-model building module is labor intensive and uses different codes (i.e. Compact source: BuildSky; Extended sources via 'Shapelets'; Diffuse Emission via Spherical Harmonics).

**7) Sky-Model Subtraction:** Once the iterative procedure of calibration-imaging has converged, the sky model (compact, extended and diffuse emission) can be subtracted from the data after applying the DD instrumental and ionospheric solutions using SageCal-CO.

**8) 21-cm Power-Spectra/Tomography:** After step 7 in principle only the (spectrally-varying) 21-cm and noise signals should remain in the residual data-set. We currently obtain power-spectra using a model that performs a Bayesian Gaussian Procession Regression analysis in the frequency direction, combined with a spatial Spherical Harmonic Analysis.

**9) Signal Inference:** The final step in the simulated and observed data analysis is to infer the parameters of the 21cm signal model.

**EOR project Hardware & Observational Data**

- **Data Processing:** The LOFAR EOR KSP team operates two GPU-based data processing clusters ('Dusk' and 'Dawn'), the latter being the most powerful. It has 1584 hyper-threaded cores (24TFLOPS; 4.6TB memory; 1.7PB internal storage; 10GB/s connectivity), distributed over 32 nodes (plus 1 server) each with four K40C NVIDIA GPUs, yielding 0.55/0.18 PFLOPS at single/double precision.

- **Data Acquisition:** Beam-formed station voltages go via glass-fibers to Groningen, where a GPU-correlator 'COBALT' generates data at 2s/3kHz, yielding 50-80TB 'raw' data in runs of 8-16hrs. These are transported to our processing clusters via a fast 10Gb/s connection. About 3100hr of data has been acquired since 2012, covering 115-189MHz.

- **Data Storage**: The 'raw' data are cleaned of interference (RFI) yielding a 3-5% data loss. The data is averaged to 2/10/10s and 12/61/180KHz, respectively. About 5PB of pre-processed data is stored on storage clusters (Groningen and ASTRON) and 5PB 'raw' data at various HPC centers: SARA (NL), Jülich (GE) and Poznan (PL).

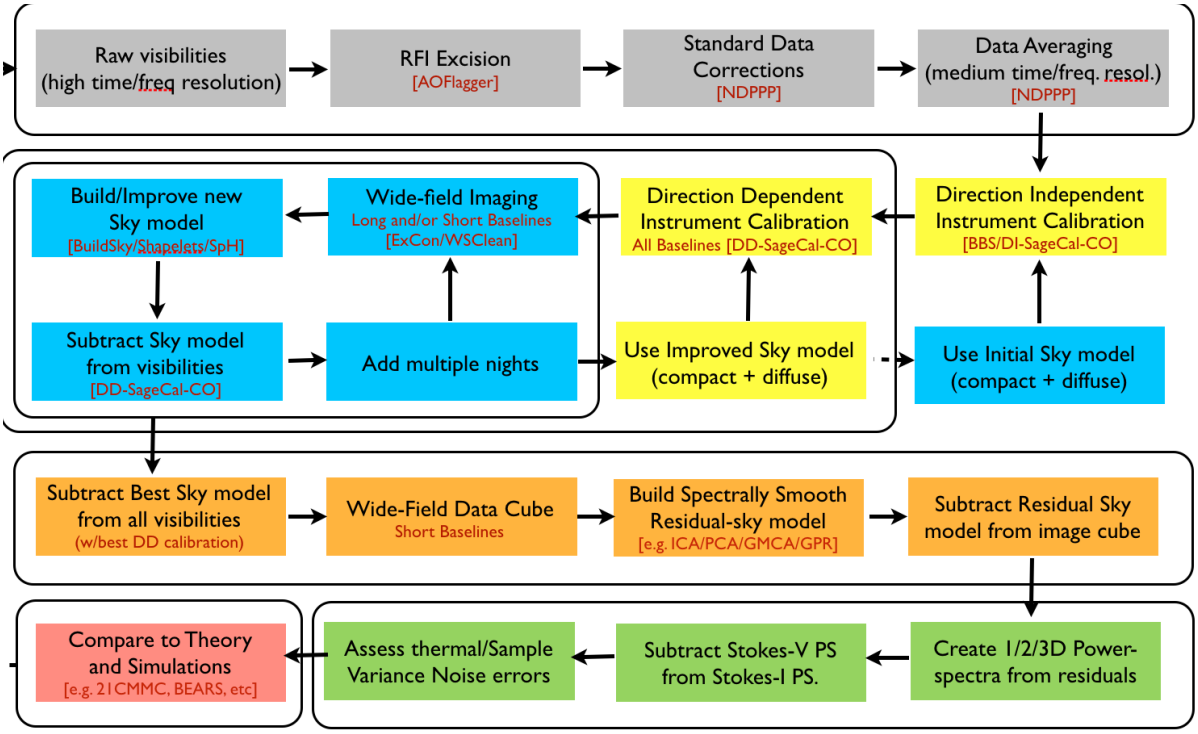



**Figure 4:** The data processing flow diagram for the LOFAR-EoR pipeline. This forms the basis for the SKA-Low forward simulation and calibration pipeline.