| Beyond Cloud Computing: Towards Complete Computing | |
|---|---|
| **HOLACloud 2015 Roadmap** (17.02.2016) | Keith G Jeffery<br>Dimosthenis Kyriazis<br>Lutz Schubert |

# Introduction

The DoA of HOLACloud included a roadmap associated with CF2016 – the Cloud Forward 2016 Conference. The project proposed and agreed to run the CF2015 conference (a) to increase the relevance of the proposed research directions of the Beyond Cloud Community; (b) to provide input to the EC from the community (focused through HOLACloud) on "Beyond CLOUD Computing" or "Complete Computing" or "I3 Computing". Thus, based on the outcomes of CF2015, the position papers submitted by the established EC clusters and the views of the HOLACloud consortium members, this roadmap has been compiled.

It should be noted that following the first HOLA Cloud conference (CF2015), a short roadmap mainly expressing the views of the authors has been produced ("Beyond Cloud Computing: Towards Complete Computing" – October 2015). The main items of this paper have been incorporated in the current version of the roadmap, while the complete paper is provided as an Annex of this document.

# Method

This paper is produced by a synthesis of opinions provided by:

1. Knowledge from the HOLACloud platform;
2. Conference papers and active discussions (especially CF2015 but also others);
3. Position papers from successful (peer-reviewed) authors at CF2015;
4. Provision of a platform for Clusters to organise themselves and provide position papers at and after, including 2 position papers that have been received from the clusters:
    a. New Approaches for Infrastructure Services
    b. Inter-cloud Challenges
    c. Software Engineering for Services and Applications
5. Active discussions at the closing session of CF2015;
6. Input from experts within the HOLACloud consortium (including chairs of the EC CLOUD Expert Group which also considered 'Beyond Cloud Computing').

These opinions cover:

1. Technical challenges of CLOUD Computing and beyond;
2. Economic and business challenges of CLOUD Computing and beyond;
3. Social and Human challenges of CLOUD Computing and beyond;

The synthesis takes account of the strength of support for particular issues and challenges based on the views of the community on the importance of a challenge, the urgency of a challenge, the feasibility of a solution to a challenge, and the economic or social benefits of such a solution.

## Rationale

The recommendations below and the draft workprogramme text are produced according to a rationale derived from the method described above.

### *Challenges Identified*

In short, the major challenges identified by the community and requiring R&D activity to improve CLOUD Computing and to move beyond conventional CLOUD Computing are:

**Policies (legal concerns, rights), Security and Privacy** including cross-border aspects consistent throughout the software mesh. This has been identified consistently as the leading challenge to uptake and further development of CLOUD Computing. The real (and ongoing) issues are:

(a) the complexity of national and international legislation – particularly concerning privacy but also rights concerning services;
(b) the risk to an organisation of having its data, software, workflows hosted at an offsite location probably replicated at other unknown or little-known locations.

**Overcoming proprietary platform lock-in**: autonomic deployment and dynamic redeployment with elastic scaling of partitioned applications, data and users across multi-clouds also dealing with (near) real-time and IoT / CPS. This has been identified consistently as the second challenge to uptake and further development of CLOUD Computing. The real issues are:

(a) standardisation of interfaces to IaaS, PaaS and SaaS offerings: to date the suppliers have refused to come to any agreement thus trying to enforce 'lock-in' to a particular proprietary platform environment to the detriment of open market competition (although the work of ETSI on CLOUD standards is progressing);
(b) overcoming the heterogeneity by middleware: providing middleware to offer a consistent interface to applications and users hiding from them the heterogeneity of the interfaces to the platform offerings;
(c) overcoming the heterogeneity by libraries: providing libraries of software for each of the offerings and for the interfaces to applications and users – a sort of 'do-it-yourself middleware kit.
(d) overcoming heterogeneity by another level of abstraction: writing an application has to become more intuitive, thereby exploiting better mapping to different platforms

**Appropriate business models for organisations utilising CLOUD Computing** to justify the switch from legacy environments and to monitor that the business (greater capability, novel applications and novel configuration of legacy applications) and computing (cost reduction, improved management control, improved compliance and governance with respect to legalistic aspects and SLAs/QoS) benefits are realised. While there are clear business benefits for an organisation to utilise CLOUD Computing for:

(a) infrequent or unusual load (computing, storage, network) requirements that would lead to increased cost with traditional models;
(b) development of software applications and testing in an environment other than the organisation production ICT environment.

It is not yet clear if there is further business benefit and in which specific cases the benefit applies – in particular transition costs and the actual impact on the software model have to be taken into consideration. While some organisations report business benefit by utilising exclusively a CLOUD Platform provider for all their ICT (effectively outsourcing it) others wish to minimise the risks concerned with legalistics and lock-in. In particular a frequent requirement is for an organisation to be able to extend its own ICT capability dynamically, elastically and on demand to one or more external CLOUD Platforms, ideally with the optimal cost-benefit.

An implicit challenge related to the three major challenges is the provision of appropriate interfaces (including to simulators as well as development/execution environments) for (1) end-users launching developed applications with appropriate feedback at different levels of detail: simple notification of launch and termination, complex (full execution trace); (2) application developers characterising the application to allow autonomic deployment where possible re-utilising knowledge gained from similar applications and their execution; (3) system administrators analysing execution monitoring information with a knowledge-based learning system to improve future deployments. There is a need for interfaces which are 'intelligent' based on knowledge engineering technologies, multimodal and multilingual, declarative rather than imperative.

Furthermore, existing CLOUD-based platform environments are (usually) not designed to deal with mobile device interfaces and mobile computing (with implicit security and sustainable communications aspects) nor with the big data five Vs: Volume, Velocity, Variety, Veracity and Value.

### *Reconsidering the Challenges*

In many ways the challenges facing 'Beyond Clouds' are those that have existed for many years in ICT namely (as above) (1) legalistics, security, privacy; (2) heterogeneity and lock-in; (c) business justification. This begs the question as to whether the existing evolution of ICT is really providing solutions to the challenges.

This was considered by (Jeffery and Schubert 2014a)[1] based on the outputs from the CLOUDs Expert Groups and Software Engineering Expert Group and further refined and elaborated in (Jeffery and Schubert 2014b)[2] and finally summarised in an invited paper for IEEE Cloud Computing[3] including the summarised roadmap. This trend of thought considers (a) the relative discrepancies of performance improvement in communications, data access and computational power, (b) recognises the 'end of the road' for processor power and the move to multiprocessor configurations, (c) notes the importance of data (and in particular the 'five Vs') and (d) reviews the failure of existing evolutionary approaches to overcoming the problems concluding by (e) challenging Turing/von Neumann architectures and proposing Information-Incentive-Intention (Triple-I) computing also known as 'complete computing' based on detailed metadata-driven knowledge processing to ensure the user

---

[1] https://ec.europa.eu/digital-agenda/events/cf/cloud-computing-software-engineering/item-display.cfm?id=14093.

[2] ec.europa.eu/newsroom/dae/document.cfm?doc_id=6775

[3] Lutz Schubert, Keith Jeffery: New Software Engineering Requirements in CLOUDs and Large Scale Systems; Invited Paper: IEEE Cloud Computing January/February 2015 pp49-57

intention is captured (i.e. abstraction of program code), the information locality, quality and utility is appreciated (abstraction of data) and the system respects the incentive (business goal abstraction) usually in the form of cost, performance, security, privacy, environmental friendliness and such parameters.

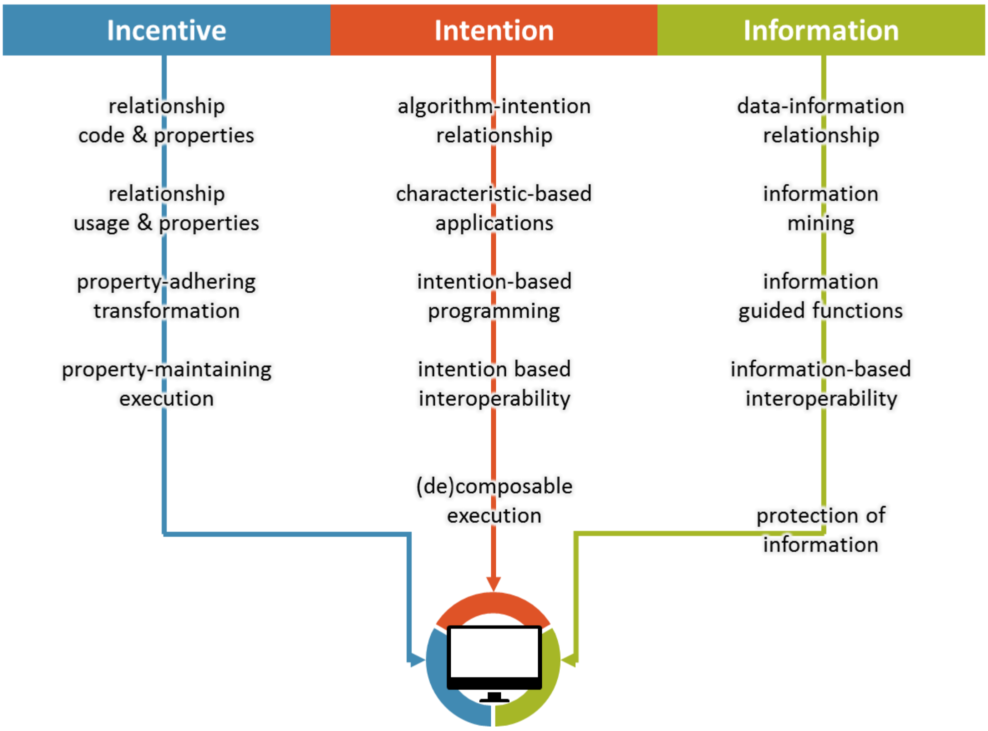
### *Two Roads Ahead*

Thus there are two ways forward:

1. Finding evolutionary novel approaches to tackle the major challenges outlined namely: (1) policies (legalistics), security, privacy; (2) heterogeneity and lock-in; (3) business justification;
2. Taking a fresh view on the challenges with a novel approach of complete computing, being willing to take "risky" leaps forward

### *Fundamental Research*

The second road requires fundamental research into computer science. The output from the expert groups implies the following topics:




#### Information

The goal of the *information* branch is to steer the development from concrete data toward abstract structures that can be transformed into different concrete data representing different views on subsets of the information.

*Investigate the relationship between data and information*: Data currently tries to be a one-to one depiction of reality. This also affects how we represent information—by using exact data. Simulations typically try to reproduce reality, rather than extract the actual information of interest.

*Information mining*: Data analysis and data mining are about extracting information from big datasets. The full potential is only achieved once the analysis acts directly on the information itself.

*Information-guided functions*: These functions deal with and aim at specific types of information. This would, for example, enable goal-oriented queries over large datasets.

*Information interoperability*: Interoperability implies not only that information can be extracted from data and that information can be understood, but also that equivalent datasets can be generated from information.

*Information protection*: Data is no longer the source of privacy concerns, but the information gained from data. With the power of information processing increasing, new ways to protect information and individuals are required.

### Intention

The *intention* behind a piece of software is still inaccessible for any compiler and even unintelligible for most human readers. The complexity of cross-referencing, pointers, function invocations, and so on makes code generally unreadable, unless it's commented on accordingly—that is, unless additional information on the intention is provided. This branch aims to make such intention a more integral part of the programming model and software engineering processes.

*Understand the relationship between intention (such as a mathematical formula) and the different algorithmic implementation/execution choices*. This includes how data is handled and transformed, as well as which intention can be expressed algorithmically with which effects.

Because distribution, scaling, and sharing substantially change the application behaviour and, in particular, its characteristics, *define application behaviour on the basis of its implicit characteristics*, rather than having to explicitly encode any potential occurrence.

*Build a syntax* (or several) that's expressive enough and at the same time carries all the information relevant for compilation and execution.

Develop the ability to *transform the different code expressions in a controlled fashion*, just as with data. In other words, intentionally equivalent (pursuing the same goals) code can be generated from the same intention declaration.

*Adapt code* (component, module) to a specific usage context, such as the device it is to run on but based on the declared intention at model level. This avoids the many-layered inefficient softare stack problem. This applies on the lowest level of granularity, as more complex functions are actually compositions of individual low-level modules.

### Incentive

Incentive is immediately reminiscent of intention. However, whereas the intention of a code reflects the behavioural goal of the application, its incentive defines the properties it needs to fulfil during execution, including user satisfaction.

*Formal analysis of the relationship* between a given code and the properties it exposes; in particular, properties related to distribution, scaling, and sharing have only been insufficiently analyzed so far.

*Impact of usage on these properties*. The properties need to be formally assignable to application characteristics so as to assess when to change the behaviour, and vice versa, in order to identify to which degree such properties can actually be fulfilled during execution. Performance and related characteristics still won't be accurately assessable, though (halting problem). Instead, it is expected that more general predictions (such as whether hosting the data somewhere else is likely to increase response time) can be made.

*Ability to transform code and data* to meet different non-functional objectives (such as quality of experience) whilst adhering to the properties.

*Bring all of these advances together* to realize composition and decomposition of functionalities, characteristics, and properties. This requires not only a full scope of transformation capabilities, but also a full understanding of all relationships of all listed aspects between each other. Transformations thereby imply composition and decomposition of functions and their properties.

*Execute software that maintains the essential incentives* (such as business goals) in a distributed, shared, and dynamic environment.

## *Evolutionary Advances*

### Base Research

The challenges above can be abstracted to basic research challenges in ICT. Given that CLOUD computing is increasingly considered as a utility, the main research challenges refer to the interdependencies:

(i) between CCD: computation, communications (i.e. networking paradigms and infrastructures) and data (across the complete data path / lifecycle);
(ii) between applications (including diverse domains such as Industry 4.0 / FoF, IoT & CPS, etc) and CCD;
(iii) between systems / software engineering and CCD;
(iv) between systems / software engineering and applications.

The requirement is for methods, models, languages and supporting tools to characterise these interdependencies that affect both the functional and the non-functional properties of the corresponding domains (i.e. CCD, software engineering, applications, and systems such as industrial environments). This characterisation will aim at enabling systems / software development to provide improved components and systems including and within CLOUD computing. The main challenge refers to approaches that will allow to close the control loops at all scales, thus addressing all of the 4 cases of inter-dependencies and enabling high degrees of efficiency, quality and automation. It should be noted that identifying, understanding and controlling these dependencies and their impact, includes also the case of conflicting ones - in terms of configuration, provision and optimization in each of the corresponding domains.

### Applied research

The applied research concerns:

Development of integrated environments – including languages and tools – respecting legalistics (rights), security and privacy throughout for characterising, deploying and executing applications on top of heterogeneous infrastructures (including singe- and multi- CLOUDs as well as edge / fog computing and hybrid environments). Key is the optimization within and (most importantly) across

CCD – e.g. orchestration through network awareness / orchestration considering data stores operations.

Development of integrated environments – including languages and tools – to optimise the application design/development and execution for a given (set of) computation, communications and data environments. The optimization should address a variety of cases, ranging from workload porting and balancing across different hybrid environments implemented with different technological stacks, to quality of service, security and privacy.

Development of business / economic models – including languages and tools – to optimise (a) the application development (b) the application execution for (1) a given business environment (including market size and state, availability of credit, competitor analysis… ; (2) a given (set of) computation, communications and data environments.

### *Mapping to Application Classes*

The following table summarizes the identified challenges for fundamental and applied with respect to main application classes. The goal of the proposed mapping is to highlight the applications (not in terms of domains but in terms of classes) that will benefit from the outcomes of the aforementioned identified fundamental- and applied- related research.

| | Minimum development and time to market | Diverse and distributed resources sensitive | Highly interactive with the physical world | Data driven and / or intensive | QoS / QoE critical | Privacy and security sensitive |
|---|---|---|---|---|---|---|
| Computation, network, data management dependencies management | | ✓ | | ✓ | ✓ | |
| COMPLETE computing architectures and implementations | | ✓ | ✓ | ✓ | ✓ | |
| Programming models: COMPLETE-oriented, intuitive, interoperable | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Service-agnostic software and data abstractions and encapsulation | ✓ | ✓ | | ✓ | | |
| High availability in COMPLETE computing | | | ✓ | | ✓ | |
| COMPLETE computing resources externalization | | ✓ | | | ✓ | |
| Cross-border and data management related privacy and legislations | | | | | | ✓ |

## Draft WorkProgramme Text

Challenge: Support the evolution of CLOUD computing towards COMPLETE computing in which computation, communications and data are optimized in a holistic way, by understanding, considering and managing their inter-dependencies both during deployment and during runtime. The challenge is for COMPLETE computing to provide all the means for different application domains (e.g. Industry 4.0, CPS, IoT, etc) in order to allow them to efficiently exploit the provided COMPLETE services – with guaranteed quality of service, security and privacy. The latter highlights the need to overcome proprietary platform lock-in within and across multi-clouds, while dealing with software-related aspects ranging from software mesh inter-dependencies with COMPLETE computing environments, to software engineering approaches enabling the development of services through declarative methods and the abstract definition of interfaces to address the needs of the complete ecosystem (user, developers, administrators). Notwithstanding the evolution of CLOUD computing towards COMPLETE computing poses an additional challenge: adaptable business models that amplify value in multi-stakeholder environments.

Scope: Research and innovation actions addressing the aforementioned challenges are expected to focus on the following:

- Understanding the relationship between algorithms & data and execution properties, ranging from scaling behaviour to data usage
- Hybrid infrastructure management techniques managing and optimizing the inter-dependencies between computation, communications and data, as well as between the infrastructure and the software instances;
- Encapsulation and abstraction of data, applications and services in new structures allowing the exploitation of heterogeneous COMPLETE computing instances (e.g. embedded systems, resource-constrained devices, etc);
- Realization of COMPLETE computing through technologies addressing the diversity in the technology stacks as well as the diversity of information and operational technologies (e.g. Industry 4.0);
- Making applications and their execution (de)composable without affecting the main quality properties, dependability and high availability. This includes autonomic deployment with elastic scaling in multi-platform (including cloud and edge computing devices) and multi-provider cases;
- Quality of service, dependability and high-availability in COMPLETE computing incorporating autonomic deployment and dynamic redeployment with elastic scaling in multi-platform (including cloud and edge computing devices) and multi-provider cases;
- Externalization of resources and development of new constructs enabling transparent migration, portability and balancing of workloads in COMPLETE computing environments;
- Service-agnostic software abstractions enabling automatic, dynamic and transparent associations of infrastructure instances (including purpose-designed ones) with software resources;
- More abstract programming models that allow more intuitive and interoperable programming without loss of performance.

- Programming models tackling the wide distribution of resources in COMPLETE computing and resulting to programming and management / operational interfaces;
- Legalistics (rights), security and privacy guarantees including cross-border aspects consistent throughout the software mesh;
- Understanding the impact of big data / data mining on privacy protection and identifying means to protect the user in new ways
- Business models exploiting the emerging opportunities in dynamic COMPLETE computing environments as well as triggering the development and evolution of such environments according to emerging situations and opportunities.

Some of these topics are best addressed by fundamental research modalities while others are best addressed by R&I. In a few cases Coordination modalities are appropriate.

**Workprogramme modalities**

H2020 has the following modalities that may be applicable in 'Beyond Cloud Computing':

1. Collaborative Projects called:
    a. Research and Innovation Actions (RIA)
    b. Innovation Actions (IA)
2. Coordination and Support Actions (CSA)
3. SME Instrument (SME)
4. Pre-commercial Procurement (PCP), Public procurement for Innovative solutions (PPI)

There is potential scope for PCP actions leading to PPI especially for the provision of Cloud-based public services by government. There would be advantage in the PCP mechanism of 2 or more procuring governments which couldlead to standardisation of interfaces between government services for the citizen (including open data) and Cloud platform suppliers.

For particular Cloud or beyond-Cloud services there is scope for the use of the SME instrument, to encourage SMEs to join the marketplace for Cloud applications of general use – especially from mobile devices. However, for this to be effective there needs to be a European defined 'standard' interface to insulate the SME applications from changes in and the heterogeneity of the underlying Cloud platforms.

CSAs are helpful in coordinating research and innovation and also leading towards standardisation. The work of ETSI and partners in Cloud Standards Coordination http://csc.etsi.org/ is relevant here but there is a clear need for extension towards Complete Computing where a clear roadmap and early adoption of standards could give Europe an important leading position creating a commercial market especially for SMEs.

Finally RIA is a well-known and well-justified mechanism for producing results which can be taken through innovation to commercial benefit. In the area of Complete Computing the major need is for this kind of action in order to explore the many issues involved and to produce both prototype products and proposals for standards that can benefit European industry.

## Conclusion

The analysis of the material from the various sources within the CLOUD and 'Beyond Clouds' communities leads to two roadmaps, one (a) based on evolutionary basic and applied research and the other (b) on Fundamental Research. We propose both roads are followed to maximise the chances of the EU Member States (a) to maintain and develop CLOUD and beyond services over the next 5-10 years; (b) to 'leapfrog' the competition and place EU Member States at the forefront of the next generation of computing 'beyond CLOUDs'.

## Annex I – Preliminary Position Paper

| Beyond Cloud Computing: Towards Complete Computing (October 2015 ) | HOLA CLOUD |
|---|---|
| Keith Jeffery, Dimosthenis Kyriazis | A roadmap paper from the HOLACloud Project 20151028 |

## Introduction

This brief report is an outcome from the Cloud Forward (CF) 2015 Conference organised by the HOLACloud Project. There are four sources of input: position papers requested from authors whose papers for the conference passed peer review; initial ideas (actually confirming the first input) from the cluster meetings; views of invited speakers; views from the authors of this report as well as from other partners of HOLA Cloud project, namely Lutz Schubert and Geir Horn. The position papers were condensed into a paper (K. Jeffery, D. Kyriazis, G. Kousiouris, J. Altmann, A. Ciuffoletti, I. Maglogiannis, P. Nesi, B. Suzic, Z. Zhao, "Challenges emerging from future cloud application scenarios") and a presentation which opened discussion for all the participants representing the community and also encouraged views from the invited speakers. This validated the key R&D topics for Complete Computing: information, intention, incentive ($I^3$ computing).

## Requirements and Novel Application Areas

The emergence of the R&D topics was predicated upon existing and especially novel application areas. These challenging application areas and their requirements include:

1. Joint Collaborative Business Intelligence platforms with multiple dynamic data sources: to provide SMEs with an otherwise unaffordable knowledge-based resource for their business benefit through the addition of "sharable" data sources;
2. Software Development on the Cloud: many organisations can benefit from a CLOUD-based development environment obviating the need for expensive in-house development and encouraging shared (open source) development and integration;
3. Joint Application/Infrastructure controllability: there is a significant disconnect in non-functional aspects between the application level and platform level developments and characteristics. This is retarding significantly CLOUD take-up. There are important sub-topics:
    a. QoS / QoE critical applications on the cloud: certain applications require guaranteed QoS/QoE and as yet such guarantees cannot be given.
    b. Adaptable Parametric Applications: this involves within execution adaptation to maintain QoS/QoE and to optimize resources. The outcome is 'intelligent' application plus platform combined to maximize the business benefit for the end-user.
    c. Generic Application Templates: their provision would allow end-users to choose an appropriate workflow with connected software, data and platforms so saving significant resources.
4. Security- / privacy- intensive applications: the lack of adequate security/privacy especially across the application/platform interface is retarding significantly CLOUD take-up;

5. Mix of mobile, cloud and legacy applications: there are significant challenges in integrating the various applications / application components across platforms and again such integration will overcome resistance to CLOUD utilization.
6. Knowledge on the Cloud: to promote CLOUDs for cultural heritage providing a rich basis for further education and an underpinning for arts, humanities and social science through data collection, association and context-delivery;
7. Knowledge from the Cloud: the CLOUD can be exploited to provide a resource for lifelong learning with significant impact on creating the knowledge society and providing up-to-date information and knowledge – analysed data;

## Research Priorities

The following research priorities were condensed from those associated with the challenging applications and taking into account the well-documented factors inhibiting take-up of CLOUD Computing (or indeed any ICT infrastructure).

1. Security, Privacy, Legalistic aspects: this involves as individual topics: single sign-on and authenticated token passing – overcome lock-in; dynamic and time-limited, role based authorisation based on contextual metadata; encryption and encoding; data partitioning – static and dynamic;
2. Interoperability: this involves as topics: metadata describing data, software, users and resources; data and data standards; software services and standard APIs; dynamic description of state and availability to allow dynamic resource placement of data and software; dynamic movement of data and software to appropriate resources; cloud brokerage;
3. CLOUD Development Integration: this involves topics: shared open source code base – building from components; workflow and dynamic (re-)composition; model-driven and constraint-managed software; appropriate level of abstraction in software; cloud integration (including application, data and process integration);
4. Data Orchestration and Virtualization: this involves topics: collection, synchronization and migration of data across different data sources, data stores and data analytics points / nodes; virtualization of data enabling abstraction of data from the underlying physical containers.