Distributed Intelligence for Physical Networks:

Sensing, Data and Analytics, Control, and Platforms

Part 2: Data and Analytics

Jeffrey D. Taft, PhD

Cisco Distinguished Engineer

Chief Architect, Connected Energy Networks

Version 1.1

November 25, 2011

Abstract

Physical networks such as electric power grids that have both structure and dynamics generate huge volumes of data as the amount of instrumentation and the number of intelligent edge devices capable of generating both data and event messages increases, to the point where managing the data and extracting information from it become “big data” problems. Dealing with the big data problem for such systems requires an understanding of the data classes produced by such systems, as well as the methods that apply to managing the data.

As a part of big data management, the issue of extracting useful information also arises. With data volumes being simultaneously far too large and too complex for human comprehension, it is necessary to make use of tools that extract essential information. Further, much of the information contained in the data must be used in fully automatic systems, such as closed loop controls and other machine-to-machine applications. The analytics that perform such information extract can, with proper definition and design, be used in a distributed fashion to not only extract the needed information, but also aid in the management of the massive data flows generated in the physical system and its ICT overlay.

This paper describes the data classes arising from the sensing, measurement, and event message streams generated by system devices in large scale systems. It provides a definition of analytics that clarifies how they may be applied to large scale systems and introduces architectural patterns for distributed and hierarchical implementations of analytics for both information extraction, and large scale data management.

Data and Analytics in Ultra-large Scale Systems

Ultra-Large Scale Systems (ULS) such as smart grids and other large scale control systems lead to Big Data problems when the underlying physical systems are densely instrumented. We must understand the nature of the data involved as well as the analytics that are used to extract actionable information from the data in order to develop proper intelligent network solutions. It is not sufficient to treat data as a transport problem, rather it is necessary to have a deeper understanding of data classes and uses than has been typical for developing products and solutions for the Service Provider industry. This is due to a fundamental difference between, say, the Service Provider industry and industries such as electric power: the primary business process of a Service Provider is data transport and the key discipline is networking; for other industries involving large scale data acquisition and control, data transport is one of many key disciplines and while data transport is important, it is not the primary business process.

Providing scalability for Ultra-Large Scale Systems involves a number of criteria, the most obvious being the avoidance of communications bottlenecks. However, the scalability issue is not simply one of communication bottlenecking – it is also (and perhaps more importantly) an issue of data management, and a matter of processing capacity. Systems that use a central SCADA system or a central data repository for data collection become latency-bound as system sizes increase. This is not an issue for some applications, but for others it is detrimental or even fatal.

Analytics are routinely used to expose information buried in large volumes of complex data. However, they may also be used to decrease data volumes while preserving information. Thus they can play a dual role: information extraction and data scale management. Consequently, we consider ULS data and analytics issues jointly. We include visualization along with analytics, since under some models visualization is a form of analytic designed for the eye and mind.

Data Classes and Characteristics

This section focuses on data coming from grid devices and systems, or data associated with the grid (meta-data). Utilities have many other data sources and consequent databases and other data elements, but the rollout of smart grid devices has the most impact on change in the utility and is the focus of much of the networking that utilities must install or upgrade.

Data arising from smart grid devices and systems may be grouped into five classes. Each has its own key characteristics; an understanding of these classes is important in the development of networking solutions for electric utilities. Table 1 below describes these five key data classes.

Table 1 Grid Data Classes

Data Class	Description	Key Characteristics
Telemetry	Measurements made repetitively on power grid variables and equipment operating parameters; some of this data is used by SCADA systems	Constant volume flow rates when the data collection technique is polling; standard SCADA polling cycles are about 4 seconds, but the trend is to go faster; telemetry can involve a very large number of sensing points. Telemetry data usually comes in small packets.
Oscillography	Sample data from voltage and current waveforms;	Typically available in bursts or as files stored in the grid device, captured due to a triggering event; transferred on demand for use in various kinds of analyses; for some kind of sensing systems waveform data is acquired continuously and is consumed at or near the sensing point to generate characterization values that may be used locally or reported out (e.g. converting waveform samples to RMS voltage or current values periodically); waveform sampling may be at very high rates from some devices such as power quality monitors
Usage data	Typically meter data, although metering can occur in many forms beside residential usage meters; typically captured by time-integrating demand measurements combined with voltage to calculate real	May be acquired on time periods ranging from seconds to 30 days or more; residential metering may store data taken as often as 15 minutes, to be reported out of the meter one to three times per day

	power	
Asynchronous event messages	May be generated by any grid device that has embedded processing capability; typically generate event messages in response to some physical event; this category also includes commands generated by grid control systems and communicated to grid devices	For this class, burst behavior is a key factor, depending on the nature of the devices, the communication network may be required to handle peak bursts that are up to three orders of magnitude larger than base rates for the same devices; also, since many grid devices will typically react to the same physical event, bursting can easily become flooding as well
Meta-data	Data that is necessary to interpret other grid data or to manage grid devices and systems or grid data	Meta-data includes power grid connectivity, network and device management data, point lists, sensor calibration data, and a rather wide variety of special information, including element names, which may have high multiplicity

Power grid devices and sensors operate in one or more of five data output modes:

- Polling – a master queries the device, which response with the most recent values of the specified data points. Such data usually comes in small packets, has high priority for traffic purposes, and is sampled on a regular and frequent basis.

- Report by exception – the device pushes a data value to the master when the data changes by a specified amount. Some telemetry is collected this way, but many utilities prefer not to use this mode.

- Streaming – sensor sends a continuous stream of data, once streaming is initiated, until streaming is terminated by command or abnormal exit condition

- Interrogation of stored files – the device maintains a log or data file; upon query, it transmits the log or file to the master; example is a PQDIF or COMTRADE file from a C&I meter, power quality monitor, digital fault recorder, or remote video system DVR. The difference between this and polling as described above is that stored files are much larger than typical telemetry packets, are low priority for traffic purposes, and are requested infrequently and at essentially random intervals.

- Asynchronous event message – the grid device or system uses internal processing to detect a specific condition indicated by the data and spontaneously sends an event message to the master or any subscribing system- the message may or may not contain actual sensor data relevant to the event; the internal triggering event can be a clock signal or countdown so that the messages are sent on a regular basis, but initiated by the edge device, not a central controller, or it may be some logical condition based on actual data; report by exception could be considered as a special case of this but is usually treated separately from event messages. When data transfer push is initiated on a regular timer basis, it has characteristics similar to polled data except the point of control is distributed to the data sources, whereas in normal polled telemetry, the point of control is central and is orchestrated by the receiving system. In the asynchronous mode, each data source sends without any orchestration, unless the source nodes coordinate with each other. In some AMI systems, the meter report usage data this way, generally with some randomization of reporting intervals in an attempt to reduce data channel overload.

Polling is common in SCADA systems, but report by exception is used in some systems to reduce data volumes and therefore communication line bandwidth. Not all utilities are willing to use report by exception. Streaming is common for advanced sensors such as PMU-based wide area measurement systems (WAMS). Interrogation of stored data files is common for meters and for data loggers and grid devices that collect records (oscillography) on a triggered basis. Asynchronous event messages are becoming more common in devices that contain significant local processing and are therefore able to detect and declare events. Older smart meter system used polling to perform mass meter reads, but in newer designs the meters push the data to the head end at times selected by the meters.

Latency Hierarchy and Lifespan Classes

Data is consumed in a variety of ways and places in a power grid; most of these are not located at the enterprise data center and much grid data does not enter the data center. Some of it does not even enter the control/operations center, as it must be consumed “on the fly” in grid devices and systems. Consequently it is important to classify data according to the latency requirements of the devices, systems, or applications that use it and appropriate persistence (or actually, lack of such) must also be defined. Keep in mind that much grid data has multiple uses; in fact, it is an element of synergy that has significant impact on smart grid economics and system design (networking, data architecture, analytics) to ensure that data is used to support as many outcomes as possible. Figure 1 below illustrates the issue of latency. Latency hierarchy is a key concept in the design of both data management and analytics applications for physical networks with control systems or other real time applications.

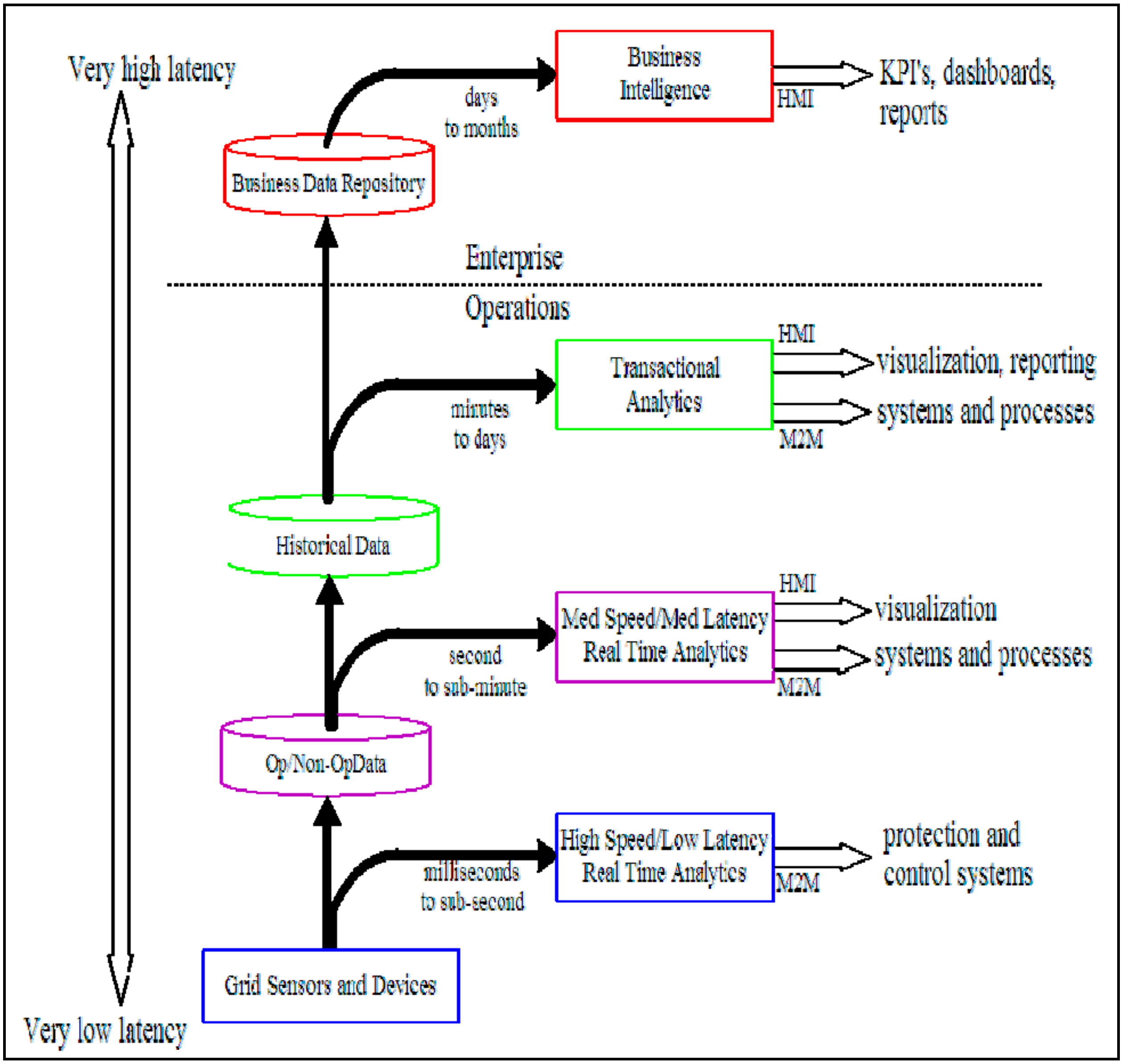


Figure 1 Latency Hierarchy for Grid Data

What the chart does not illustrate is that a given data element may in fact have multiple latency requirements, depending on the various ways it may be used, meaning that any particular datum may have multiple destinations.

The latency hierarchy issue is directly connected to the issue of lifespan classes, meaning that depending on how the data is to be used, there are various classes of storage that may have to be applied. This typically results in hierarchical data storage architecture, with different types of storage being applied at different points in the grid that correspond to the data sources and sinks, coupled with latency requirements. Table 2 below lists some types of data lifespan classes that are relevant to smart grid devices and systems.

Table 2 Grid Data Lifespan Classes

Data Lifespan Class	Characteristics
Transit	Data exists for only the time necessary to travel from source to sink and be used; it persists only momentarily in the network and the data sink and is then discarded; an examples are an event message used by protection relays, and sensor data used in closed loop controls; persistence time may be microseconds
Burst/Flow	Data that is produced in bursts or is processed in bursts may exist temporarily in FIFO queues or circular buffers until it is consumed or overwritten; examples include telemetry data and asynchronous event messages (assuming they are not logged) – often the storage for these data are incorporated directly into applications, e.g. CEP engine event buffers
Operational	Data that may be used from moment to moment but is continually updated with refreshed values so that old values are overwritten since only present (fresh) values are needed; example: grid (power) state data such as SCADA data that may be updated every few seconds
Transactional	Data that exists for an extended but not indefinite time; typically used in transaction processing and business intelligence applications; storage may be in databases incorporated into applications or in data warehouses, datamarts or business data repositories
Archival	Data that must be saved for very long (even indefinite) time periods; includes meter usage data (e.g. seven years), PMU data at ISO/RTO's (several years); log files. Note that some data may be retained in multiple copies; for example, ISO's must retain PMU data in quadruplicate.

Just as with latency hierarchy, grid data may progress through various lifetime classes as it is used in different ways. This implies that some data will migrate from one type of data storage to another as its lifetime class changes, based on how it is used.

Prioritization of Grid Data Access

At the operational level, there is a definite hierarchy of data access priority. These are naturally based on functional use of the data. In order from highest priority to lowest, at the power system control and field levels, they are:

1. Protection
2. Transmission Control

3. Alarms and Device Event Messages
4. WAMS Data
5. Distribution Control
6. Security Data (IM, RBAC, etc)
7. Secondary Load Monitoring (i.e. demand response data)
8. Telemetry (i.e. remote asset monitoring)
9. Usage Data (metering)
10. Voice/video/data for field force

Specific exceptions to this prioritization can exist on a case by case basis, so that the actual prioritization must be determined for the data traffic on any specific utility network.

Primary Data Framework Components

Many types of storage and database technologies are useful in the smart grid context. Table 3 below summarizes principal types. Some types are specialized for specific purposes or applications; others like standard SQL databases are used in more general applications.

Table 3 Storage Types

Store Type	Comments
Operational Data Stores	Used to hold state data which is continually refreshed, such as power and device state data, real time grid topology
Time Series Stores	Used to hold telemetry that will be processed in various ways over various time scales, but specifically including very long times
FIFO Queues and Circular Buffers	Very short term storage for data being consumed quickly by applications; often implemented in the application itself as memory resident small volume buffers
Meter Usage Data Repositories	Large scale repositories for meter data; these often hold the data of record for billing; generally associated with meter data management systems, although some independent MUDR's have been implemented
Relational Databases	Widely used in a variety of operational and enterprise contexts; built using either standard relational database technologies or memory-resident versions for faster response, especially in business intelligence and decision support applications. Utilities may have many such databases that have grown organically over many years of operation.
Warehouses and Datamarts	Used for storage of very large data sets for business intelligence, data mining, and the like; generally relational, but newer approaches are emerging

True Distributed Databases	Databases in which various data elements exist in non-duplicated form on various physical stores, non-duplication being key to scalability; useful for operational data/grid state in distributed intelligence environments
Waveform Repositories	Used to hold waveform files (oscillography); the waveform files may be treated as BLOB's; repositories can be special purpose or a general content management tool
GIS as a Data Store	Geographic Information Systems are often the system of record for as-built physical network topology (occasionally it may be the Outage Management System that performs this function for Distribution); some smart grid applications need access to the as-built topology meta-data, so it can be necessary to use the GIS as a database, although most are not built for real time or near real time query support. Consequently, as-built topology may be staged to a datamart for near real time access, with periodic updates form the GIS to the datamart.
Federated Databases	This is not a database type so much as a middleware for databases; federation can tie together heterogeneous databases so that querying systems do not need the details of the multiple underlying databases; this technology, along with CIM-structured relational databases has been used to integrate multiple operational, transactional, and time-series databases in smart grid data management solutions
No-SQL/No-ReL databases	Developing in response to "big data" requirements, these databases avoid the use of relational structure (hence the names "No-SQL" and "NoRel"), these databases are intended to scale to petabytes and beyond. These are beginning to see some use for business intelligence applications but have not penetrated utilities much as of this writing (Nov 2011).
Content Manager Stores	Databases designed specifically for content management, so that files of various kinds can be stored, access-controlled, version-controlled, etc. Useful for BLOB-like objects, hence the mention above for waveform repositories, but also useful for engineering drawings, video, manuals, and grid device settings/configurations

For more specifics on data for power systems, see the white paper "Extended Grid State and Sensing Strategy for Smart Grids v 6" *(Extended Grid State and Observability v6.doc).*

Analytics and Visualization

Data analytics are tools (normally software) that extract essential information from masses of data. The term is quite general, and so refers to many different type of data processing. We include visualization along with analytics because in most cases we need to present the results of the analytic processing in ways that are easily comprehended. In fact, in some models, visualizations are considered to be analytics specifically crafted for the eye and brain. Generally, analytics are used to provide input to decision and control processes.

We subscribe to a flexible definition of analytics that does not depend on specific software tools or processing techniques such as data mining or Business Intelligence. In our definition, an analytic is a data processing algorithm implementation that decreases Shannon entropy. This definition allows us to view low level signal processing operations such as noise filtering as analytics, in that they reduce uncertainty in the information content of a data record or stream (for streams, we may use entropy rate). This definition also makes it clear that exactly reversible transformations such as Fast Fourier Transform (FFT) or symmetric components are in fact not analytics by themselves. Such a result agrees with intuition (the time domain and frequency domain representations of a signal are exactly equivalent, as are unbalanced phasors and their symmetric components) and clears up a common misstatement about FFT being an analytic. Such transformations are often very useful for making the information extraction process easier to accomplish and therefore will often be incorporated into analytics as preprocessing steps.

The definition also provides a view of a chain of analytical data processing steps as creating an entropy pyramid: as we move up the processing chain, entropy should decrease at each stage. If we consider the analytic as an information channel, then we may apply the concept of transinformation to the analytic, and we then recognize that an analytic may be able to reduce entropy, thereby permitting a reduction in data volume or data rate. Consequently, analytics can serve not only to extract useful information from data, but they can also be used as data volume reduction tools, especially when applied in a hierarchy or processing pipeline Basically, if a raw data record has a given entropy per bit, and if the analytic is information-lossless, then the output can be represented with fewer bits than the input so that we can keep entropy per bit roughly constant as we progress through the analytic chain and up the entropy pyramid.

For electric utilities and their power grids, there are a wide variety of useful analytics. These range from very low level signal analytics (primarily digital signal processing) to power system performance metrics, to consumer behavior analytics. Figure 2 below illustrates a taxonomy of analytics for smart grids. Based on the foregoing definition and discussion, it is clear that low level signal processing operations can be analytics; hence the category of signal analytics. In other areas, we find more traditional KPI’s, metrics, and figures of merit, as well as the outputs of event correlation, pattern classification, and various detection processes.

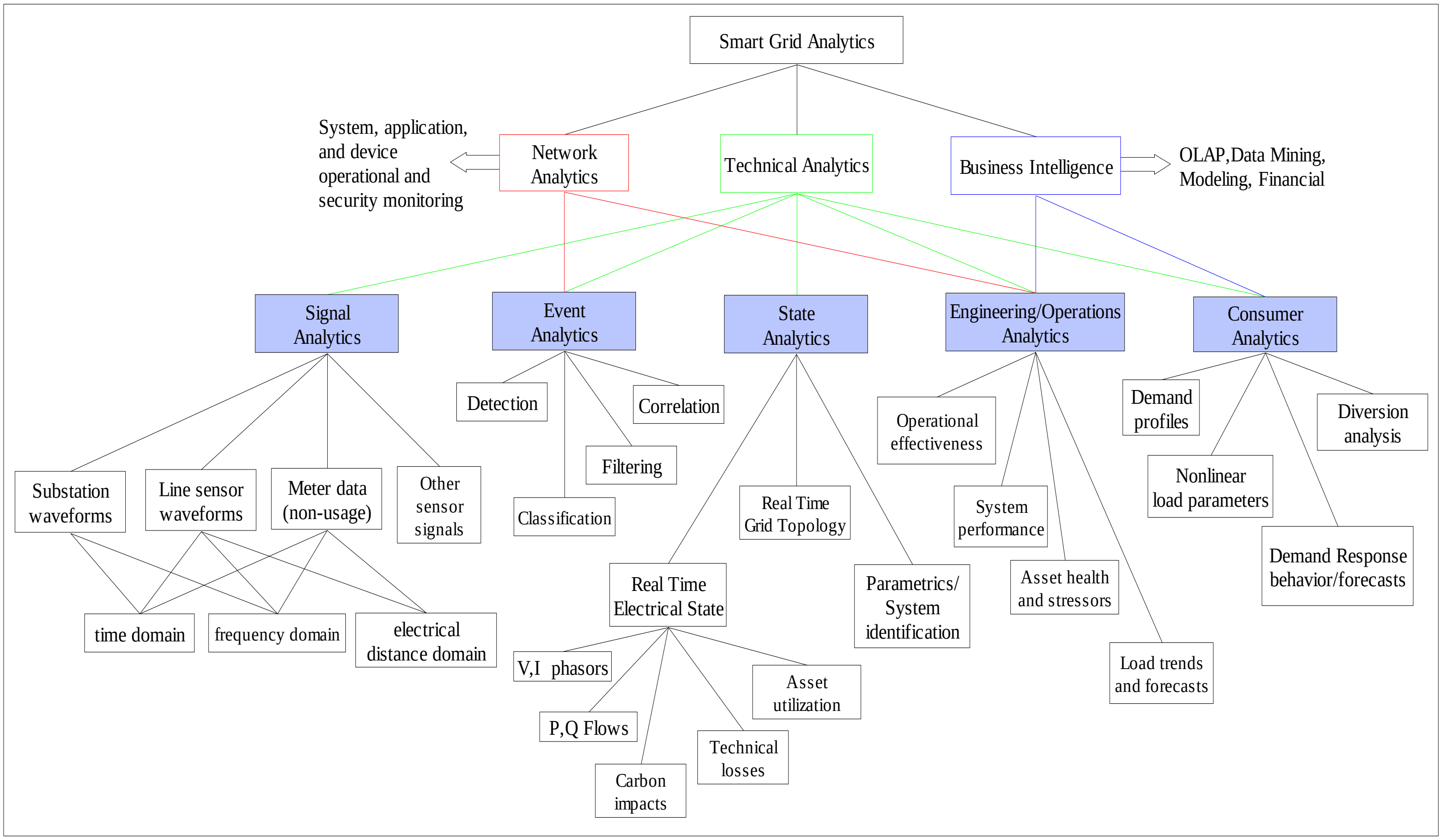
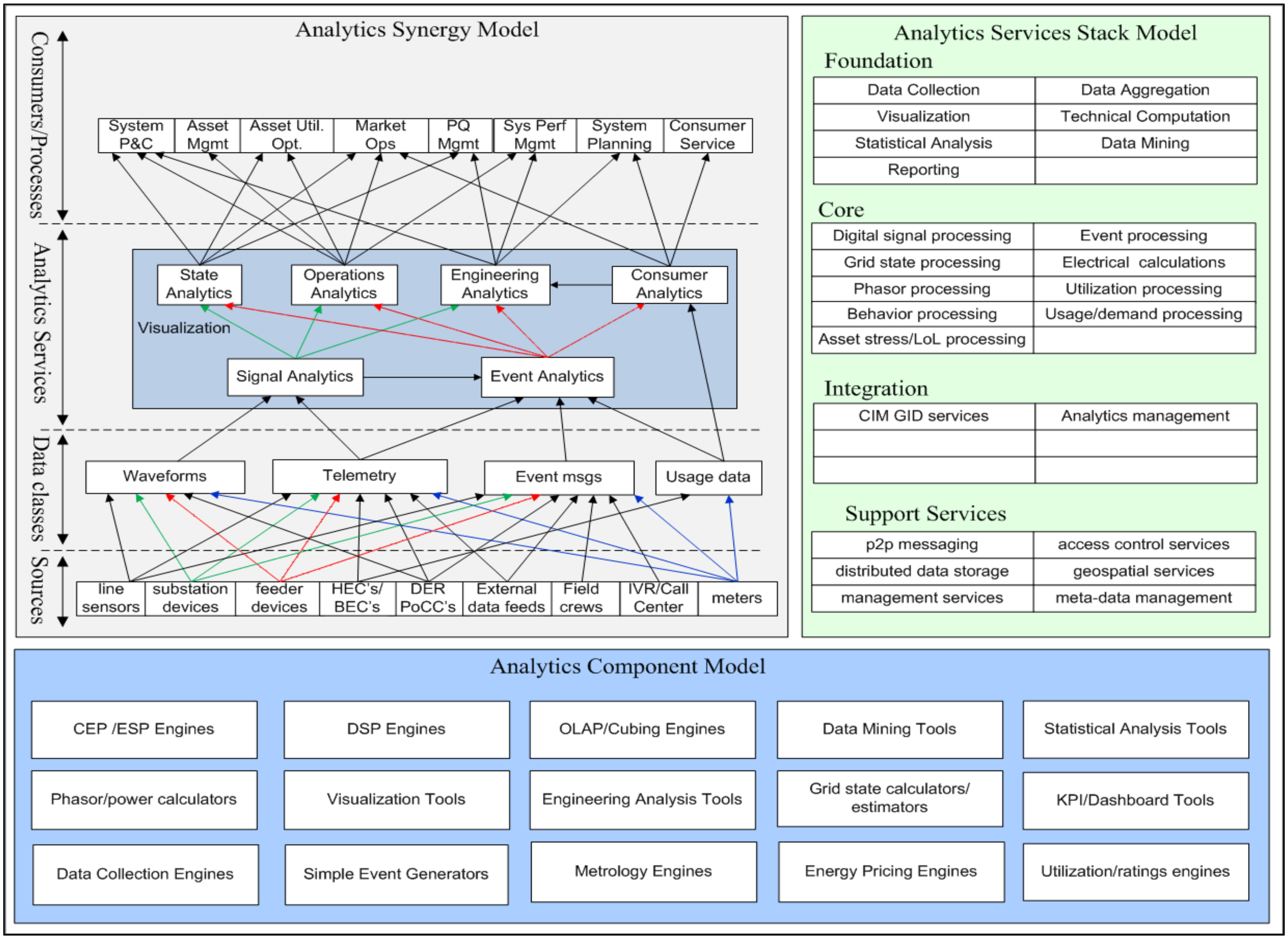


Figure 2 Power Grid Analytics Taxonomy

In any complex system, such as s smart grid, the set of useful analytics can be quite large, and many different analytics may be applied to the same data in order to support multiple capabilities and outcomes. In fact, one of the design problems for smart grid is to select the set of analytics that maximize the useful information extracted across all desired outcomes, from a minimized set of sensors. Such designs relay upon a core set of analytics components, as well as foundation services for processing, integration, and support. To design a robust and efficient analytics system, one must understand the synergies that can be achieved across analytics classes, as well as the relationships to data classes and capabilities classes. Figure 3 below illustrates this set of abstract relationships for smart grids.



Analytics Services Stack Model

Foundation

Data Collection	Data Aggregation
Visualization	Technical Computation
Statistical Analysis	Data Mining
Reporting	

Core

Digital signal processing	Event processing
Grid state processing	Electrical calculations
Phasor processing	Utilization processing
Behavior processing	Usage/demand processing
Asset stress/LoL processing	

Integration

CIM GID services	Analytics management

Support Services

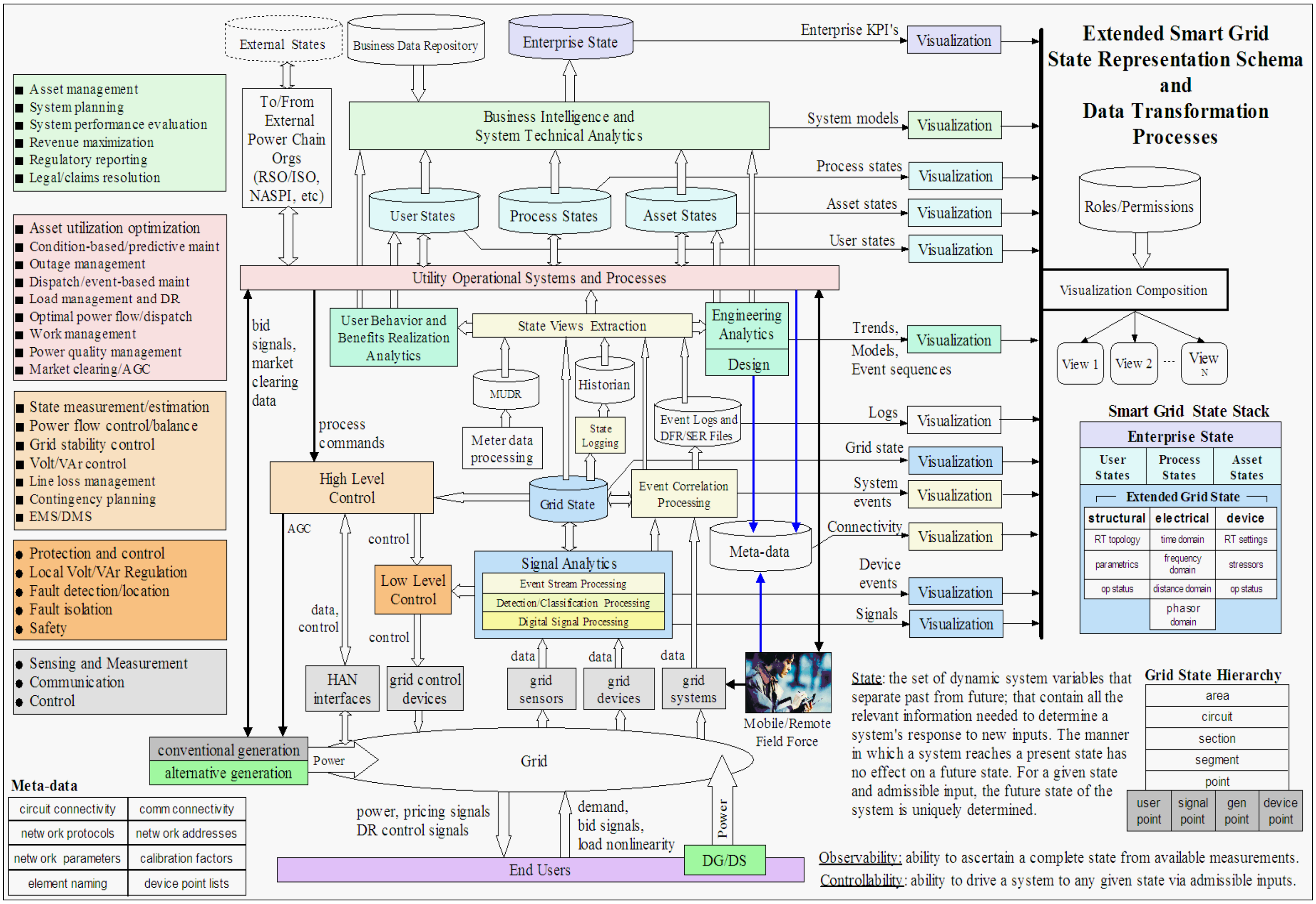
p2p messaging	access control services
distributed data storage	geospatial services
management services	meta-data management

Analytics Component Model

CEP /ESP Engines	DSP Engines	OLAP/Cubing Engines	Data Mining Tools	Statistical Analysis Tools
Phasor/power calculators	Visualization Tools	Engineering Analysis Tools	Grid state calculators/ estimators	KPI/Dashboard Tools
Data Collection Engines	Simple Event Generators	Metrology Engines	Energy Pricing Engines	Utilization/ratings engines

Figure 3 Smart Grid Analytics Component, Service and Synergy Models

Visualization is an integral part of any smart grid analytics system, and it seems likely that this holds true for any large complex system. Applying the concept of system state, we can see how to integrate many levels of visualization with the data acquisition system. Applying hierarchical layering, we can arrive at a multi-layer visualization approach that resembles generalized version of the layers model used by many geographic information systems. In such a model, compositing of logical visualization layers can provide for very flexible visualization solutions. Figure 4 shows an example of how to integrate layered visualization with extended grid state and control processes.



Meta-data

circuit connectivity	comm connectivity
netw ork protocols	netw ork addresses
netw ork parameters	calibration factors
element naming	device point lists

Smart Grid State Stack

Enterprise State		
User States	Process States	Asset States
Extended Grid State		
structural	electrical	device
RT topology	time domain	RT settings
parametrics	frequency domain	stressors
op status	distance domain	op status
	phasor domain	

Grid State Hierarchy

area			
circuit			
section			
segment			
point			
user point	signal point	gen point	device point

State: the set of dynamic system variables that separate past from future; that contain all the relevant information needed to determine a system's response to new inputs. The manner in which a system reaches a present state has no effect on a future state. For a given state and admissible input, the future state of the system is uniquely determined.

Observability: ability to ascertain a complete state from available measurements.

Controllability: ability to drive a system to any given state via admissible inputs.

Figure 4 Layered Visualization

Analytics may be implemented in a fully centralized manner, such us usually done with Business Intelligence tools, which operate on a very large business data repository. However, for real time systems, a more distributed approach may be useful in avoiding the inevitable bottlenecking. A tool that is particularly suited to processing two classes of smart grid data (streaming telemetry and asynchronous event messages) is Complex Event Processing (CEP) which has lately also been called streaming database processing. CEP and its single stream predecessor Event Stream Processing (ESP) can be arranged into a hierarchical distributed processing architecture that efficiently reduces data volumes while preserving essential information embodies in multiple data streams. Figure 5 shows an example of such analytics architecture. In this case, the analytics process line sensor data and meter events for fault and outage intelligence.

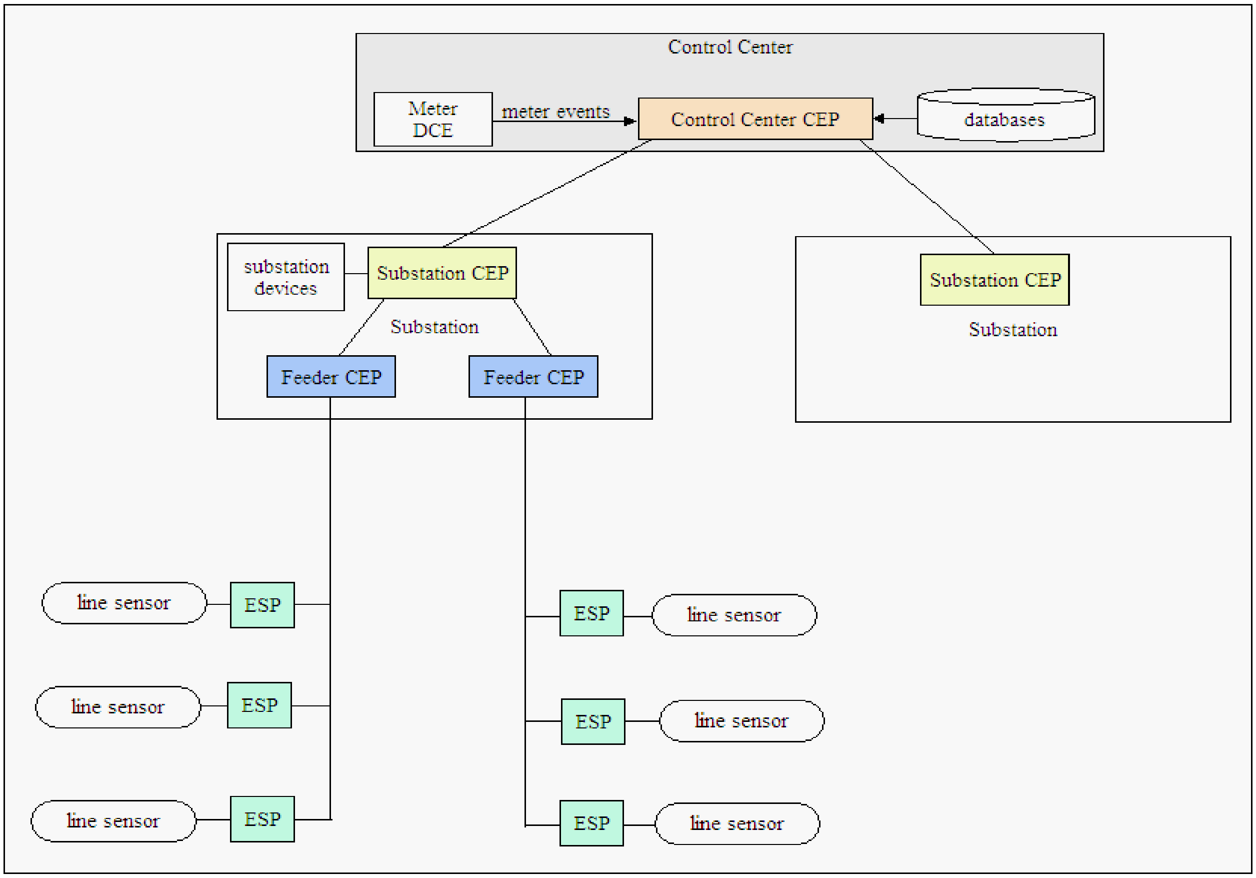


Figure 5 Hierarchical CEP/ESP Analytics Architecture

In general, distributed analytics can be decomposed in to a limited set of analytic computing elements, with logical connections to other such elements. Full distributed analytics can be constructed by composing or interconnecting basic analytic elements as needed. We have defined five basic types of distributed analytic elements:

1. Local loop – an analytic element operates on data reports its final result to a consuming application such as a low latency control

2. Upload – an analytic element operates on data and then reports out its final result

3. Hierarchical – two or more analytic elements operate on data to produce partial analytics results which are then fused by a higher level analytics element, which reports the result

4. Peer to peer – two or more analytics elements operate on data to create partial results; they then exchange partial results to compute final result and each one reports its unique final analytic

5. Database access – an analytic element retrieves data from a data store in addition to local data; it operates on both to produce a result which can be stored in the data store or reported to an application or another analytic element

Figure 6 illustrates these five basic types plus a generic node synthesized from the five basic prototypes.

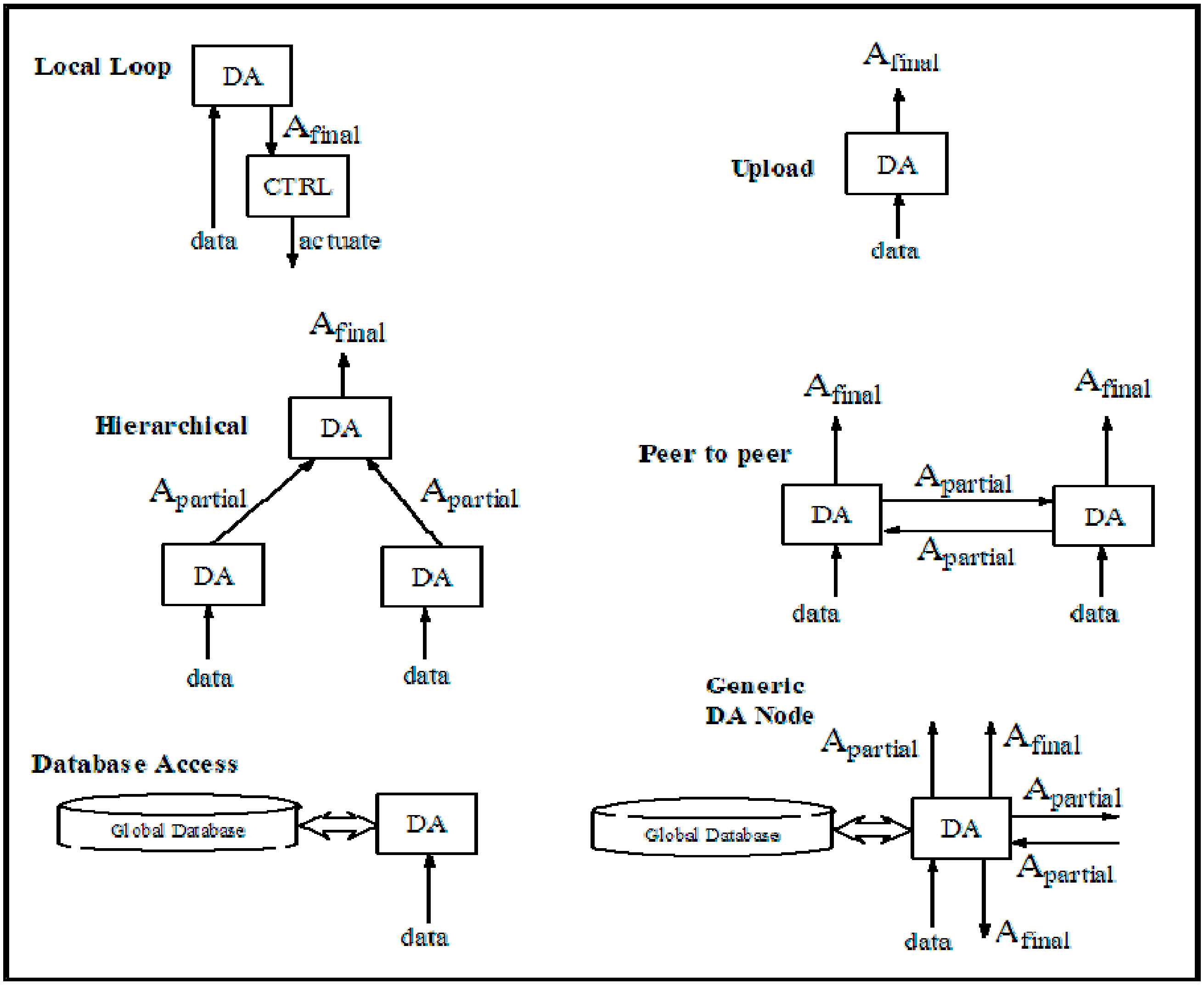


Figure 6 Distributed Analytic Basic Element Types

Distributed Data Persistence

Given the introduction of the concept of distributed analytics, including the database access element shown in Figure 6, it becomes useful to consider distributed data persistence as an architectural element. Low level and low latency analytics for smart grids (mostly related to control) require state information and while local state components are always needed, it is often the case that elements of global state are also necessary. Operational data (essentially extended system state) may be persisted in a distributed operational data store. The reason for considering a true distributed data store is for scalability and robustness in the face of potential network fragmentation. In power systems, it is already common practice to implement distributed time series (historian) databases at the control center and primary substation levels. We may incorporate this and the distributed operational data store into an integrated data architecture by employing data federation in conjunction with various data stores. Figure 7 below illustrates a data store architecture that federates distributed and centralized elements in order to support a wide range of analytics, controls, and decision support for business processes.

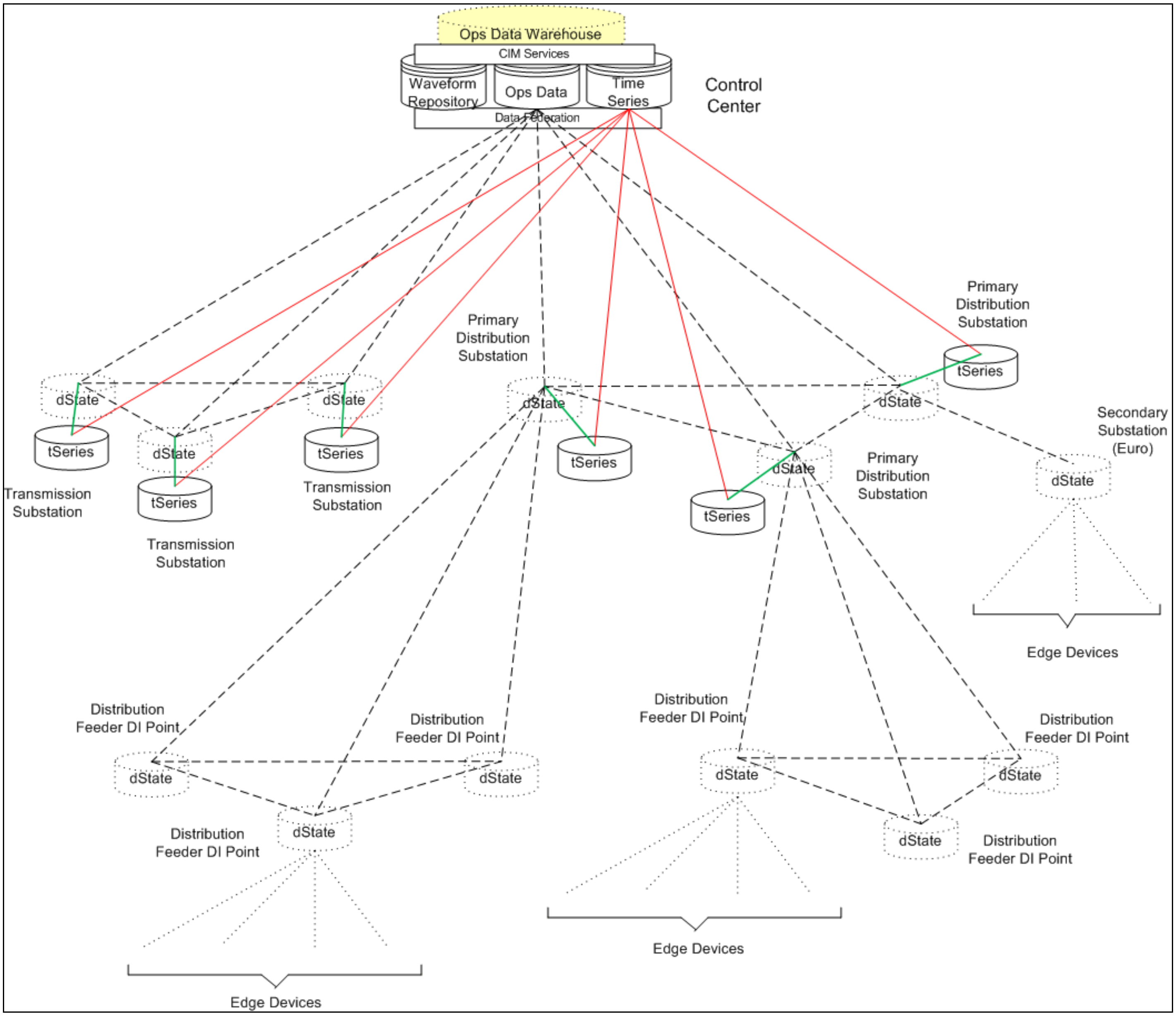

Ops Data Warehouse
CIM Services
Waveform Repository
Ops Data
Time Series
Data Federation
Control Center
Primary Distribution Substation
Primary Distribution Substation
Primary Distribution Substation
Secondary Substation (Euro)
dState
tSeries
Transmission Substation
Transmission Substation
Transmission Substation
Distribution Feeder DI Point
Distribution Feeder DI Point
Distribution Feeder DI Point
Distribution Feeder DI Point
Distribution Feeder DI Point
Distribution Feeder DI Point
Edge Devices
Edge Devices
Edge Devices

Figure 7 Integrated Data Store Architecture for Smart Grids

Conclusion

Ultra Large Scale Systems that have underlying physical networks and dynamics, especially those needing control inherently incorporate several classes of data flows. The characteristics of these classes are key requirements for system architecture. As the scale of smart grids and similar large scale control-oriented systems increases, the volumes of data expand beyond the abilities and capacities of traditional data management systems. A variety of data management tools must be brought into play, including distributed data stores, as part of a hierarchy of data persistence tools.

Analytics, needed to extract actionable information from masses of data, must be considered in conjunction with data classes and data management issues in the design of ULS solutions. The importance of analytics is two-fold: first and foremost as the means to interpret “big data” and secondly as a means to perform data management for large scale data flows.

With a proper definition of data management and analytics, the task of managing big data, such as the data associated with smart grids and other physical systems, is strengthened considerably. The ability to view analytics as a processing chain or hierarchy leads to the concept of distributed analytics, as well as distributed data persistence.