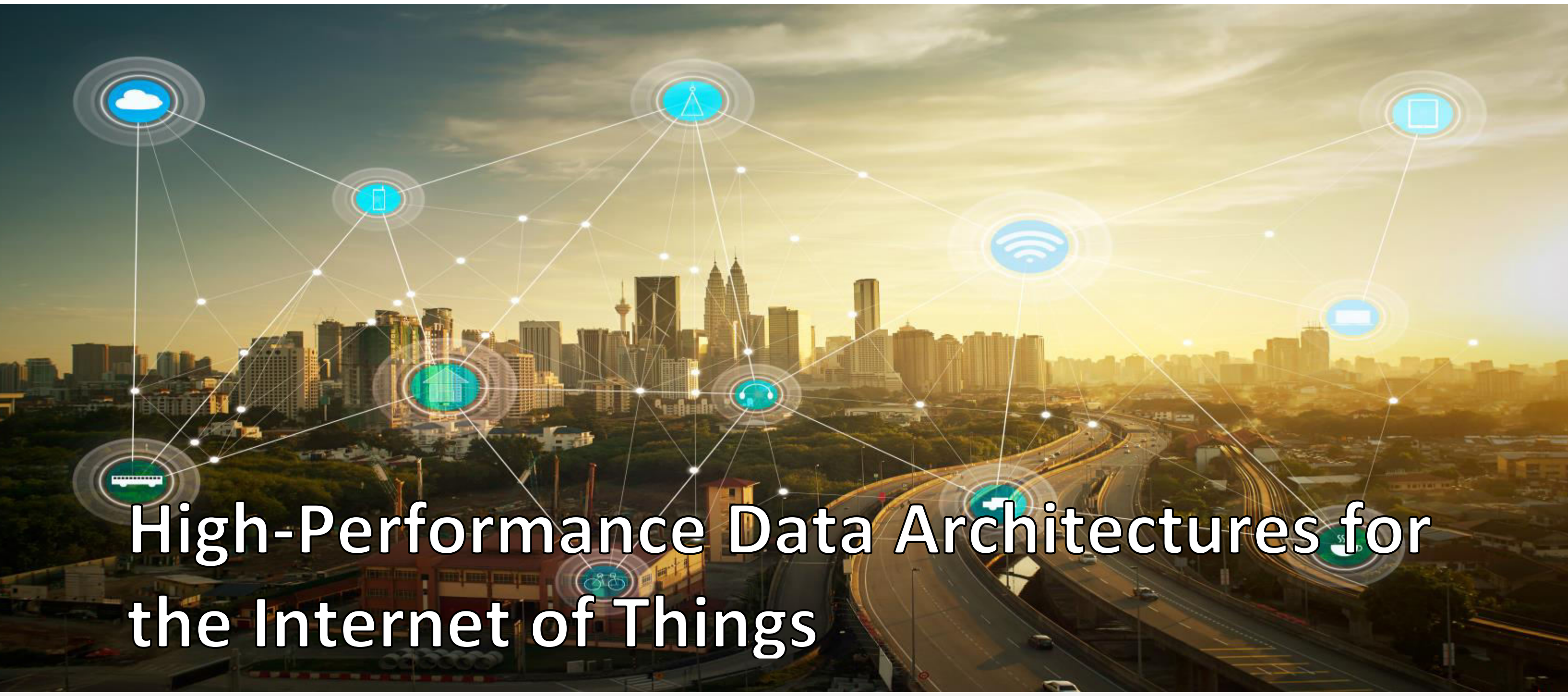
WHITE PAPER

GridGain



High-Performance Data Architectures for the Internet of Things

A GridGain Systems In-Memory Computing White Paper

January 2017

Contents

In recent years, the number of smart, connected devices making up the Internet of Things (IoT) has grown explosively. In 2009, when Cisco estimates the IoT was born, people were just beginning to play with micro-controllers to build smart devices. Since then, the number of devices has grown rapidly, and the IoT has expanded to include not just traditional computer-based devices and electronic appliances, but also wearables — such as watches, clothing, and rings — and vehicles, from the smart, self-driving cars currently in development to drones and other flying things.

By 2020, Gartner expects the IoT to have over 20 billion connected things. With that many connected devices transmitting information, there will be an enormous amount of processing to be done. To cope with this rapid expansion in the Internet of Things, successful IoT platforms will need a data architecture that can address significant challenges in terms of speed, scalability, variable workloads, and other issues.

What type of data architecture can handle these challenges? Before discussing the technology needed to tackle all of the issues around IoT, let's take a closer look at some popular IoT use cases.

IoT Use Cases and Opportunities

According to 451 Research, 65% of companies are using IoT and 69% of organizations gathering data from end points while 94% of the companies that collect that data use it for business purposes. The highest usage is among Utilities (92%) and Manufacturing (77%).

The following use cases are among the most popular:

- **Smart homes** – smart meters and home automation
- **Wearables** – clothes, watches, findable keys, virtual and augmented reality devices
- **Manufacturing, retail, and industrial** – sensory tracking for production line monitoring
- **Transportation** – intelligent traffic control and self-driving vehicles

We are seeing these types of use cases, described in more detail below, with many of our customers.

Smart homes. Smart meters are already implemented in the U.S., and British Gas is working toward government targets of having smart meters in all U.K. businesses and homes by 2020. Many companies are also working on home automation systems, with use cases emerging around thermostats and other sensing devices, as well as robotics and tracking. Applications are being developed around pets, kitchen appliances, and gardens, as well as other areas of the home.

Wearables. The Android and Apple watches were early entries in this category. Smart clothes are also in the works, with details of how they will interact still being worked out. Another major use case involves Bluetooth key rings and other types of tags that let users tag physical items and track them in a digital world.

The wearables category also includes devices such as headsets that allow users to explore virtual or augmented reality. Facebook's substantial investment in Oculus, maker of the Oculus Rift headset, helped convince other investors and developers to focus on virtual reality — triggering substantial innovation in new sensors, new human-computer interaction devices, and new ways to map physical reality to virtual reality.

Manufacturing. Using sensors to track production lines is becoming a useful way to streamline industrial processes. This type of tracking is already happening in car manufacturing and other industries.

One example of how sensors have been helpful is in the process of oil drilling. Every time an oil drill breaks, it causes a very expensive disruption for the oil industry — many workers must stand by while the broken drill is removed and a new one is installed. Sensors provided a way for this industry to track and better understand the drilling process. Analysis of data from the sensors improved the industry's ability to predict when drills were going to break.

Transportation. With both ride-sharing companies and taxi drivers using apps to broadcast anonymous information about their travel, there's a lot of data becoming available for intelligent traffic control. Finding a better route through traffic is a very popular use case that is likely to complement the expanding development of self-driving vehicles by Google, Mercedes, and other companies.

The Challenges of IoT

The types of opportunities described here don't come without challenges, including the following:

- **Scale -** Billions of connected devices
- **Speed -** High-frequency operations and transactions
- **Distribution -** Operability from any place at any time
- **Security -** Privacy protection
- **Omni-channel –** Connection through any device and any path (omni-session)
- **Variable workloads -** Analytical and transactional

Let's look at these challenges in more detail.

Scale. As noted earlier, Gartner is predicting 20 billion connected devices by 2020. The ability to take information from so many devices and derive useful meaning from this data is very challenging.

Speed. If companies scale up but can't process the information fast enough to inform decision-making, the effort is pointless. In the example of the oil drilling sensors, if the analytics aren't processed in time to prevent drill breakage, then the information is too late. Similarly, with traffic controls, if the information isn't received quickly enough, it's probably no longer accurate — and likely to do more harm than good.

Distribution. In the Internet of Things, connected items must remain connected at all times from any location. Addressing geographical distribution will require strategies such as multi-site appointment. Services will need to be dynamic and have the ability to cater for such scenarios.

Security. With the large number of connected sensors and other devices, there will be a lot of personal information being relayed at all times — information about people's houses, locations, health, and so on. Having the ability to secure this information and offer effective privacy protection will be extremely important.

Omni-channel. This term originated in e-commerce — referring to the ability to perform bank transactions through multiple contact channels, including ATMs and the web — but it's also relevant to how the overall IoT needs to work. Whether people are using phones, watches, laptops, cars, or other devices, they should be able to go through similar paths and end up in the same backend, if they're executing similar types of operations.

Variable workloads. The ability to execute hybrid workloads, both analytical and transactional, is one of the key challenges in IoT. Being able to execute transactional operations, such as getting money from an ATM, is very important. It's also important to use analytical processing to ask interesting questions about the data — for example, using traffic data to predict travel times or using sensors in an oil drilling process to predict when drills are going to break. The success of IoT will depend on the ability to execute both types of operations in one system.

The need to solve these challenges has shaped the development of a specific type of architecture for IoT.

Basic Components of the IoT

Before taking an in-depth look at the architecture for the Internet of Things, let's start with its simplest building blocks. Basically, the IoT is made up of the following:

- **Things** (a.k.a., devices) that can communicate digitally: for example, mobile phones, cars, and a wide-range of items tagged with sensors or built around small, single-board processors or microcontrollers (for example, Raspberry Pi, BeagleBone, or Arduino)
- **Communication mechanisms** through which the devices communicate with backend servers either directly, in the case of intelligent devices communicating through Wi-Fi or Ethernet, or indirectly, as when readers gather data from Bluetooth devices or devices with NFC or RFID tags and send the data through a gateway
- **A server-side Infrastructure** — high-speed, distributed, and cloud-based — for processing the data from these devices and deriving meaning from it, as well as for providing data storage and access to users

Now, let's look a little deeper at the software components of the server-side infrastructure, and see how they fit into this picture.

Figure 1 - IoT Reference Architecture

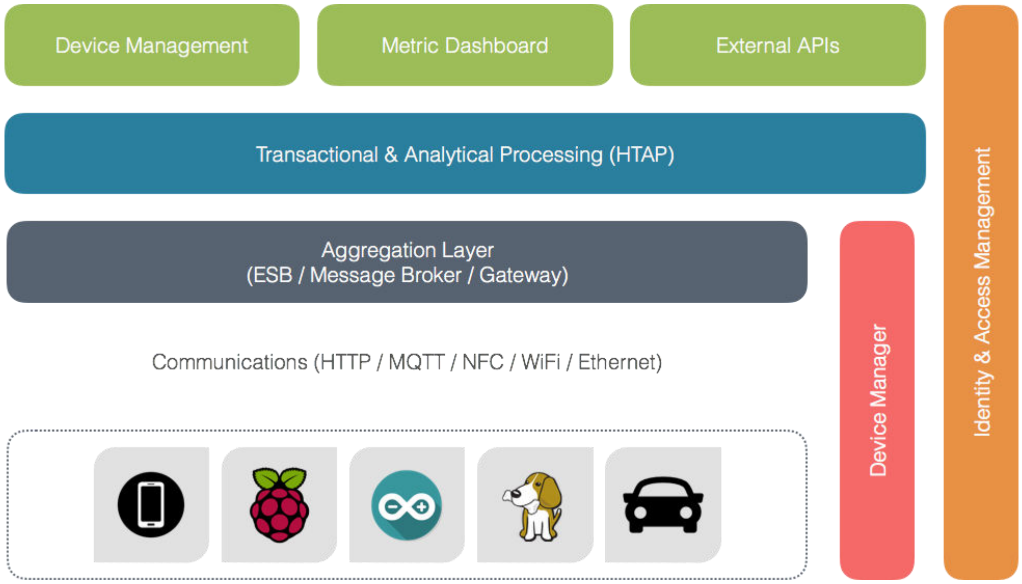


Figure 1 shows the things (devices) and the communication mechanisms in the lower-left hand corner, with infrastructure components wrapping around them along the top and at the right. Components closest to the outside of the diagram are closest to users.

The components of the IoT infrastructure software include the following:

- **Aggregation layer.** Messages can be tracked or replayed through this layer to provide resilience. This layer might consist of an Enterprise Service Bus (ESB), a message broker (such as RabbitMQ or Kafka), or some type of gateway implementation.

- **Transactional and analytical processing layer.** This layer provides hybrid transaction and analytical processing (HTAP) for handling both transactional and analytical workloads. The ability to handle both types of workloads is a key IoT requirement that we will discuss in greater detail in the next section.

* **Device management.** This user-interface component gives users access to devices for activities such as running maintenance and checks.

- **Metric dashboard**. This user-interface component lets users run analytics (such as predictive analyses) and batch operations on the device data.

- **External APIs**. This interface component makes data available for third-party systems to use.

- **Device manager**. This layer allows administrators to manage physical devices (deployments, updates, and so on).

- **Identity and access management.** This security-oriented administration component defines who can access devices and any related services and operations.

The most crucial of these components with respect to the overall success of the IoT is the layer that provides hybrid transactional and analytical processing — the HTAP layer. This layer has two important roles: facilitate time-sensitive transactions and create meaning from the massive quantity of IoT-generated data.

The Architecture of HTAP for IoT

The architecture of the HTAP component of IoT is similar to the Lambda architecture defined by Nathan Marz for Big Data applications, and it takes advantage of both stream- and batch-processing methods. The Lambda architecture, shown in Figure 2a, includes the following two layers:

- **A high-speed layer** — a real-time processing and transactional engine (typically something like a caching system and a compute grid, such as Redis and Spark)

- **A batch/storage layer**— data storage with an analytical or historical processing engine, such as Hadoop with Hive

We can apply the concepts in the Lambda architecture to the HTAP component of the Internet of Things because the HTAP component deals with event-stream processing, fast analytics, and storing data for advanced and long-term historical analysis, when necessary.

Comparing Figures 2a and 2b, we can see what the Lambda architecture looks like in an IoT context.

Figure 2a - Lambda Architecture

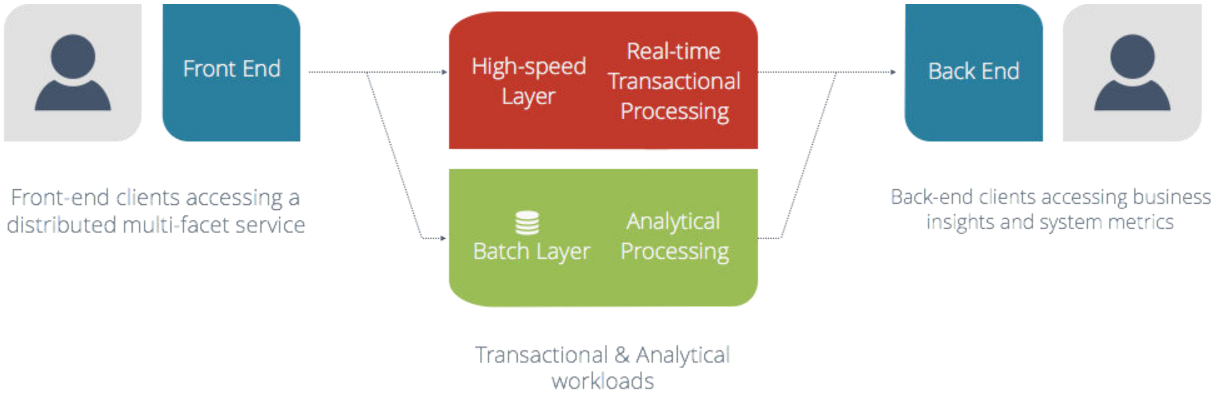
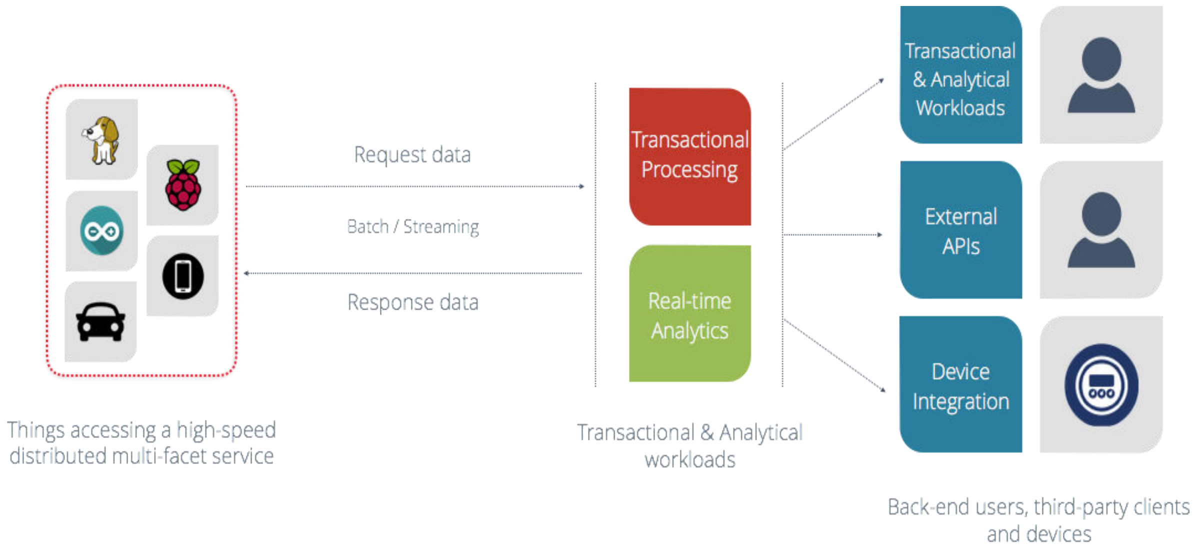


Figure 2b - Lambda Architecture for IoT



In the IoT version of the Lambda architecture (Figure 2b), the devices communicate either batch data or streaming data through a message bus to an HTAP layer for transactional and analytical workloads. The HTAP layer then provides fast analytics and transactional processing to backend users, third-party clients, and other devices.

The obvious next question is: How do we bring this architecture to life and make it work?

Multiple HTAP Technologies vs. One Solution

One way to enable the type of hybrid transaction and analytical processing needed for IoT is to stitch up a number of different existing technologies. For example, the SMACK stack combines Apache Spark, Mesos, Akka, Cassandra, and Kafka for this purpose: Kafka as the messaging layer for streaming data, Mesos to manage clusters for fast analytics, Cassandra as a distributed database, Akka for event processing, and Spark as the overall compute framework and processing engine.

For companies just getting started with IoT, this approach of combining multiple technologies can require a significant investment in terms of skill set. While it is possible to find people who know each of these technologies, it's not so easy to find people who know *all* of them. There is a lot of complexity involved.

In contrast, the GridGain in-memory computing platform provides a way to simplify this architecture. It handles the needs of both transactional and analytical processing, as well as providing persistency and event processing — all in a high-speed, linearly scalable platform. And, of course, it's just one core technology with one skill set to learn, instead of several.

The GridGain In-Memory Computing Platform: One Solution with Integrated Modules

From a big-picture standpoint, the GridGain In-Memory Data Fabric is a high-performance, distributed, in-memory platform for computing and transacting on large-scale data sets in near real-time. Taking a closer look, it is a collection of modules: in-memory components that solve the problems of providing high performance and scalability.

The key modules of the GridGain in-memory computing platform with respect to IoT use cases include:

- Data grid
- Compute grid
- SQL grid
- Service grid
- Streaming
- Advanced clustering

Because the GridGain Professional and Enterprise Editions are the enterprise versions of Apache Ignite, all of these modules have been developed and tested as part of the entire platform. They start out already integrated and aligned to provide high-performance, distributed computing.

Let’s look at these modules in a more detail.

Data grid. Originally conceived as a simple cache sitting on top of a database, the in-memory data grid is a distributed hash map that is able to store objects, or maps. In essence, it's a key value store that can be queried. Storing data in memory makes reads and writes as fast as possible.

Compute grid. A stateless grid that provides high-performance computation in memory using clusters of computers and parallel processing, the compute grid complements the data grid. It can quickly access data from the data grid — an important capability when the data isn’t being shipped with the job and a call to a displaced database would slow things down. The compute grid can work with that data immediately, performing complex calculations or quantitative analytics and writing the results back into the data grid.

SQL grid. The in-memory SQL grid provides in-memory distributed database capabilities that are ANSI SQL-99 compliant, fault tolerant, and horizontally scalable. It fully supports all SQL and DML commands and can be accessed using standard SQL through the GridGain JDBC and ODBC APIs to provide true cross-platform connectivity from all languages. The system supports free-form SQL queries, distributed SQL joins, and cross-cache joins allowing the entire in-memory computing platform to perform like an in-memory SQL database.

Service grid. A service grid provides the ability to run in-memory services. Rather than having a traditional service-oriented architecture, with a service implementation on the server side and an interface for clients and consumers, grid service instances are deployed across the distributed data grid and compute grid that are running at the same time.

Streaming. The streaming module provides the ability to take in as much data as possible — that is, an endless stream of information coming in — and process it in real-time. That scenario is exactly what needs to happen in IoT situations.

Advanced clustering. GridGain nodes can automatically discover each other, so there is no need to restart the entire cluster when adding new nodes. Developers can also leverage the hybrid cloud support in Apache Ignite, which enables connections between private clouds and public ones, such as Amazon Web Services – thus providing the best of both worlds.

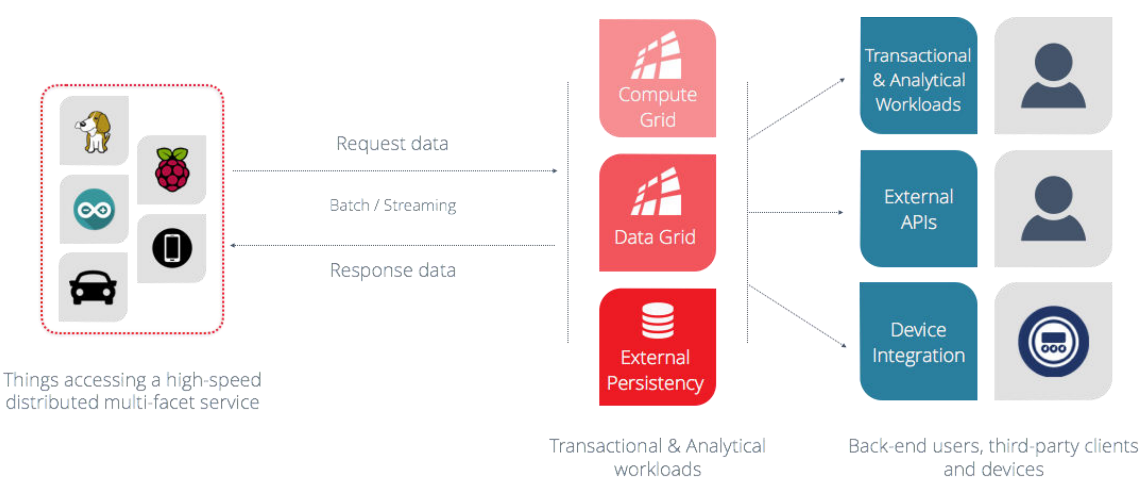
These interconnected modules are deployed as a layer between the client applications and incoming or outgoing data, including any data that may be stored on an external persistency layer (GridGain supports this option). Basically, these modules form a thin layer running in memory for greater speed — the type of speed needed to deal with the torrent of data flowing from the Internet of Things.

GridGain: A High Performance Data Architecture for IoT

Let's step back and see how the architecture of the GridGain in-memory computing platform fits into the bigger picture of IoT.

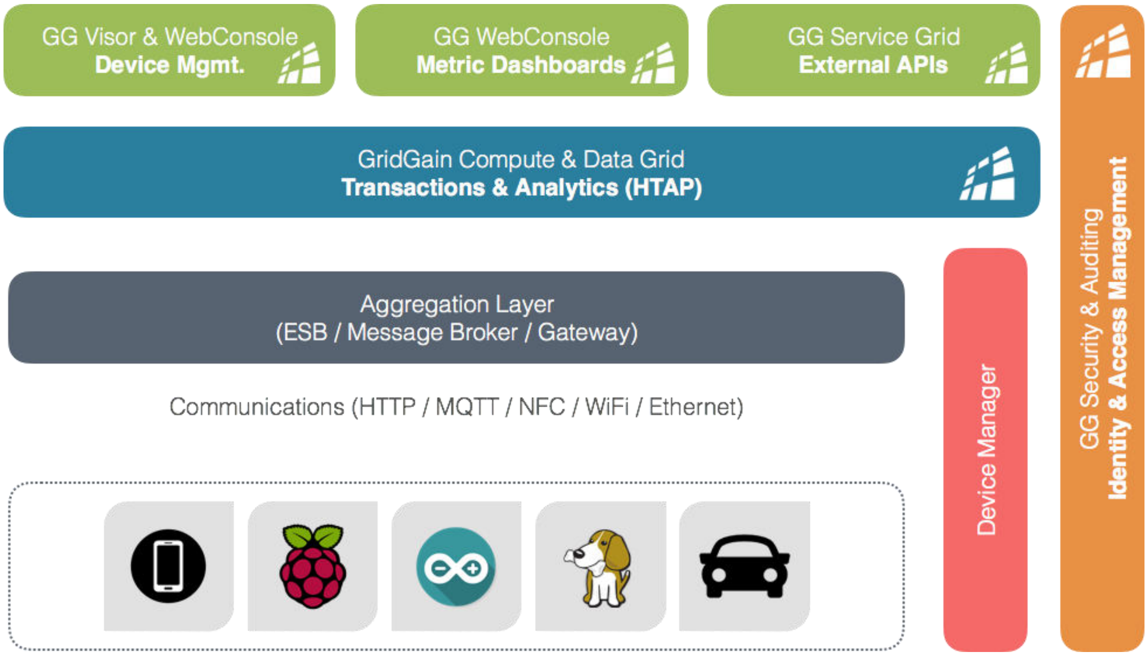
Figure 3 shows the Lambda architecture for IoT, with components of the GridGain in-memory computing platform performing the HTAP functions. The GridGain compute grid and data grid provide transactional processing and real-time analytics, while the GridGain streaming module (not shown here) also plays an important role in handling the high volume of streaming IoT data.

Figure 3 – GridGain HTAP for IoT



Revisiting the IoT architecture from Figure 1, we can see how other aspects of the GridGain in-memory computing platform fit into that architecture, as shown in Figure 4.

Figure 4 – IoT Reference Architecture with GridGain



Looking outward from the central HTAP portion of the architecture, handled by the GridGain compute and data grids, we see that the GridGain service grid supports the "External APIs" component, which makes IoT data available to third-party systems. In addition, two GridGain GUI tools – GridGain Visor and GridGain Web Console – provide user access for managing devices and for running analytics and batch operations, filling in the "Device Management" and "Metric Dashboards" components. GridGain also includes advanced security and auditing tools for the "Identity & Access Management" piece of the IoT architecture; these tools are essential for meeting the challenge of privacy protection in a scenario with devices relaying large amounts of personal information.

By implementing these core pieces of the IoT architecture in a fully integrated, linearly scalable, high-performance data fabric, GridGain delivers a platform that can ably handle the challenges of next-generation IoT. The integrated modules in GridGain combine to deliver speed, scale, distribution, security, and the variable-workload HTAP processing that is critical to IoT success.

Summary

The Internet of Things is characterized by large numbers of devices delivering massive amounts of streaming data. Lambda architectures can provide the capabilities required to create hybrid transactional/analytical processing (HTAP) platforms that are capable of providing real-time insights into the massive amount of streaming data to power useful actions at the device level. The GridGain in-memory computing platform can simplify the deployment of a Lambda architecture with a full featured HTAP architecture included in a single solution.

Contact GridGain

To learn more about how GridGain In-Memory Data Fabric can help your business, please email our sales team at sales@gridgain.com, call us at +1 (650) 241-2281 (US) or +44 (0) 7775 835 770 (Europe), or complete our contact form to have us contact you.

About GridGain Systems

GridGain is revolutionizing real-time data access and processing by offering an enterprise-grade in-memory computing platform built on Apache® Ignite™. GridGain solutions are used by global enterprises in financial, fintech, software/SaaS, ecommerce, retail, online business services, healthcare, telecom and other major sectors. GridGain solutions connect data stores (SQL, NoSQL, and Apache Hadoop) with cloud-scale applications and enable massive data throughput and ultra-low latencies across a scalable cluster of commodity servers. A converged data platform, the GridGain In-Memory Data Fabric offers the most comprehensive, enterprise-grade in-memory computing solution for high-volume transactions, real-time analytics and hybrid data processing. The company is funded by Almaz Capital, MoneyTime Ventures and RTP Ventures. For more information, visit gridgain.com.