ThingLogix Foundry And The Rise of Serverless Architecture

Perspectives from ThingLogix, Inc.

By Carl Krupitzer, Rob Rastovich, and John Mack

September 2017

Understanding Serverless Computing

The history of computing in the modern era is characterized by a repeating pattern of growth, pause, and resurgent growth. Technical breakthroughs drive early adoption, which gathers momentum that fuels an initial wave of mass adoption and hyperscale growth. As the newly adopted technologies become more mature, breakthrough innovation yields to incremental refinements, causing technical capabilities to evolve more slowly and economic growth to moderate. Eventually, another technical breakthrough emerges, either as a more efficient version of a prior technology or as an entirely new technological approach to the challenge at hand. Within the realm of computing, we are witnessing the early stages of an important phase in the evolution of cloud computing – the emergence of serverless computing.

The history of cloud computing is well known among technologists and technology investors. Single-location models yielded to locally distributed models, locally distributed models evolved into large-scale distributed models, and as this occurred, ownership of the underlying infrastructure – network, compute, and storage – no longer implied that the owner of the data was necessarily the owner of the infrastructure. Cloud computing, broadly encompassing SaaS, PaaS, and IaaS, initially bifurcated into private and public cloud models. It has since evolved into complex hybrid models that allow organizations the flexibility to dynamically manage workloads in a manner that conforms to specific preferences regarding performance, security, compliance, avoidance of vendor concentration, or other factors.

Serverless computing and the related concept of Function-as-a-Service (FaaS) represent an important extension of the cloud computing model, insofar as they significantly enhance the technical capabilities of systems and business processes that rely upon them, create market conditions for dramatically accelerated business adoption of cloud technologies, and enable new business models, higher velocity of innovation, and stronger business results.

What are serverless computing and FaaS? Three definitions can help frame the conversation.

Per Wikipedia:
Serverless computing is a cloud computing execution model in which the cloud provider dynamically manages the allocation of machine resources. Pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity. It is a form of utility computing.

Serverless computing still requires servers. The name "serverless computing" is used because the server management and capacity planning decisions are completely hidden from the developer or operator. Serverless code can be used in conjunction with code deployed in traditional styles, such as microservices. Alternatively, applications can be written to be purely serverless and use no provisioned services at all.

Per Gartner:
Serverless computing uses cloud attributes natively to execute user-defined functions, giving the consumer limited control over the infrastructure. Serverless computing solutions execute logic in environments with no visible VM or OS.

Per Stackify:
FaaS is the concept of serverless computing via serverless architectures. Software developers can leverage this to deploy an individual "function", action, or piece of business logic. They are expected to start within milliseconds and process individual requests and then the process ends.

Principles of FaaS:

- *Complete abstraction of servers away from the developer*
- *Billing based on consumption and executions, not server instance sizes*
- *Services that are event-driven and instantaneously scalable*

ThingLogix has embraced both the solution architecture principles and the business value proposition of serverless computing and incorporated them into our cloud-based platform and products for smart, connected solutions.

ThingLogix: Optimizing IoT Solutions in The Serverless IT Environment

ThingLogix offers Internet of Things (IoT) solutions, solution components, and services based on serverless architectures. Our market model is best described as "Product-led, Platform-enabled, Services-supported."

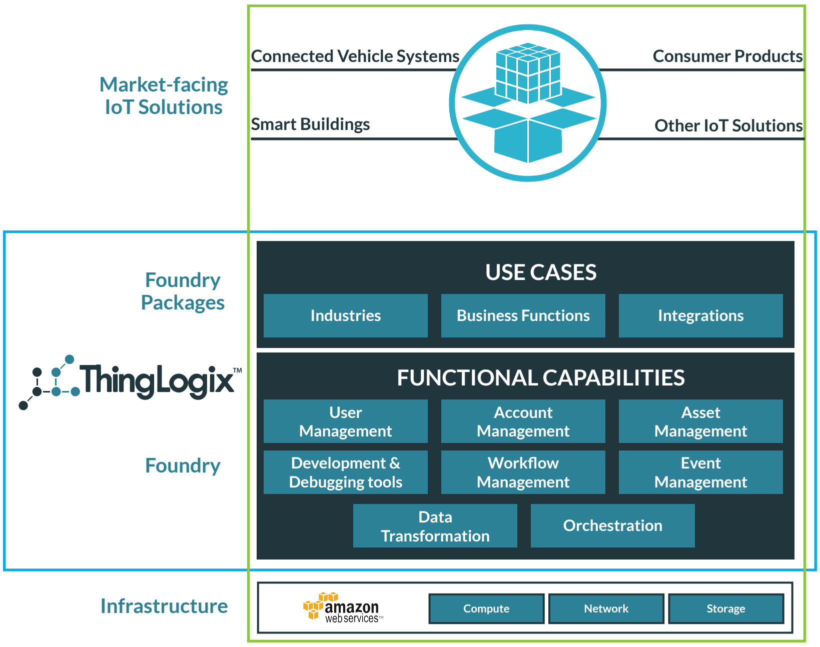
Our offerings include:

THE PRODUCT: Foundry Packages
Sets of composite IoT functionality packaged to enable specific use cases.

THE PLATFORM: Foundry
A cloud-based solution development and management platform (PaaS) that offers critical core functionality for managing the entire lifecycle of IoT solutions.

THE SERVICES

- Development services: development of IoT solutions based on Foundry, Foundry Packages, and AWS serverless infrastructure and services
- Advisory services: helping companies optimize both the business and technology dimensions of IoT solutions



Our technology enables sophisticated functionality, rapid time-to-market, simplified solution evolution over time, and extreme cost efficiency. Foundry Packages enable market-facing IoT solutions for specific industries and use cases, supporting specialization, differentiation, and competitive advantage. Foundry optimizes the development, deployment, and operation of market-facing IoT solutions architected on Amazon Web Services (AWS).

Our development services take a full-lifecycle approach, from business planning, to design and build, to testing and refining, to deploying and integrating, and on to operating, maintaining, and evolving the solution over time. For our development services clients, we architect IoT solutions on Foundry, which simplifies our clients' ability to manage and evolve their solutions over time. By abstracting the complexity of the underlying cloud architecture and simplifying solution management workflows, Foundry lets developers and product teams focus on running their business, not on writing and rewriting code over the life of the solution.

With a history of success developing, implementing, and managing IoT strategies and solutions in a wide variety of complex enterprise settings, we believe that we offer our solution customers and professional services clients a level of credibility that is distinct in the market.

From its inception, ThingLogix developed Foundry as a serverless PaaS with a focus on IoT. We have architected Foundry to align with the enterprise's increasing shift to serverless computing by embracing the concept of reusable mini- and microservices. According to industry analyst Gartner, "Miniservices and microservices are highly complementary service models. Monolithic applications are refactored into shared, reusable miniservices that reduce the scope of a service down to an individual capability. Miniservices are designed to support composition and reuse. These services publish APIs that can be accessed from client apps and from other services, and they enable integration and interoperability across application systems."

Foundry alleviates the underlying architectural complexity associated with serverless by implementing a flexible framework that allows developers to use configuration instead of custom code to quickly develop business processes on top of the core microservices needed to orchestrate underlying compute resources. This dramatically improves application development cycle time, as well as time associated with maintaining and evolving fielded applications over time as the underlying cloud infrastructure-level services change. It also renders the question "where do I even put my code" completely obsolete. Additionally, the flexible framework upon which Foundry is based allows them to easily package Foundry application functionality for distribution. This helps businesses strengthen their ability to create a robust ecosystem of developers, offer highly specialized vertical applications and, enable a marketplace for pay-per-use microservices.

Foundry Is An Event-driven Application Platform

Digital transformation is a term that broadly describes the integration of digital technologies into business models, processes, and market offerings in such a way that a company conducts business operations in a radically different, more agile, and more efficient manner. As more companies embrace digital transformation, they will encounter a key aspect of this wave of change: the shift from data-centric drivers of decisions and actions to event-centric drivers. This means that the simple occurrence of predefined events will increasingly drive business models, direct business processes, and power both physical and software-centric market offerings. ThingLogix believes that this shift to a central role for event processing in digital business will drive broader adoption of serverless computing. This, in turn, will benefit serverless-centric PaaS providers and companies using cloud platforms architected on serverless principles.

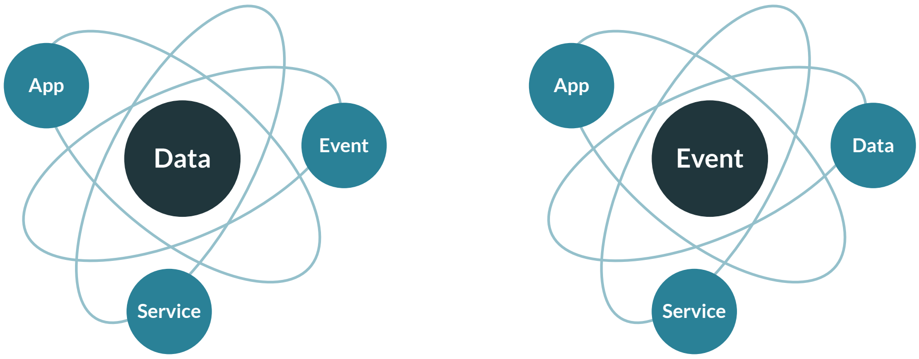
In the past, enterprise platforms have often struggled to gain widespread adoption. Reasons for these challenges vary widely, and include such issues as:

- Perceptions that the platform represented a non-essential middleware layer in the technology stack
- Platforms' often narrow focus on a specific use case (e.g., sales, mobile, a specific developer language)
- Customer aversion to relying too heavily on a specific vendor (i.e. concern about "vendor lock-in")
- An implicit requirement that the customer would need to adopt a fundamentally different, and often unproven, technology approach
- A preference to wait for market validation before adopting (i.e. assessing whether the platform has emerged as a viable industry standard vs. remaining an obscure alternative, failing to gain adoption, and lapsing into obscurity)

The Internet of Things – a network of devices and systems operating in synchronicity to achieve purposeful, coordinated outcomes – is a concept that has increasingly become a reality, and that stands to significantly accelerate companies' evolution from data-centric to event-driven businesses. When internetworked devices communicate with each other and with other systems, the devices can take action to effect change in the environment in which the device operates. Devices essentially become intelligent sensors and actuators, with the ability to take action autonomously or after prompting the regulating system for permission to act and receiving such permission. The "currency" that flows through such an internetworked system is events. The system designer defines how a given device should react in response to specific types of events, and when such events occur, they serve as the catalyst to trigger whatever actions the device has been programmed to take.

Event-driven solution architectures have profound implications for how businesses gather information, make decisions, take actions, and compete in the marketplace. Industry analyst Gartner has described how IT architectures will undergo a fundamental shift from data-centric ecosystems to event-centric ecosystems. (see accompanying diagram)

DATA STORE-DRIVEN: THE OLD WAY	EVENT-DRIVEN: THE NEW WAY



The source of truth is the data store.
First priority: Preserve data

The source of truth is the log of events.
First priority: React to events

SOURCE: GARTNER (OCTOBER 2016)

Data-centric models rely on the collection of data in a data store, which ingests event information, informs applications, and directs actions to occur in the form of services. Event-based architectures place events at the center of the architecture, such that the event itself is the information that triggers actions (services), informs applications, and flows to the data store. The event-based architecture accelerates the cycle time of workflows by predefining actions that reduce or obviate the need for data analysis in response to an event. Event-based architectures also complement machine learning and artificial intelligence technologies, insofar as these technologies allow systems to learn from a series of events, identify patterns, formulate decisions, and either recommend action or take action autonomously.

The digital technology platform underlies an organization's approach to managing business models, processes, and market offerings. As such, it is important to consider the aforementioned shift through the lens of the platform.

There are five major types of digital technology platform in the enterprise:

- **Information system platforms** — Support "back office" operations such as ERP and other core systems, as well as associated middleware and development capabilities that deliver solutions.
- **Customer experience platforms** — Contain the principal customer-facing functionality, such as customer and citizen portals, multichannel commerce, and customer applications.
- **Analytics and intelligence platforms** — Provide information management and analytical functionality. Includes data management programs and analytical applications that enable data-driven decision making, as well as algorithms that automate discovery and action.
- **Business ecosystem platforms** — Support the creation of, and connection to, external ecosystems, marketplaces and communities. The main elements are API management, control, and security.
- **IoT platforms** – Collections of technologies and standards that form a base set of capabilities for communicating with, controlling, managing, and securing elements of internetworked devices and systems.

The event-driven nature of IoT will increasingly drive the centralization of events in enterprise platforms overall, not merely in IoT-centric platforms. This shift to event-driven architectures is due to the fundamental improvement in event-to-action cycle time that such architectures represent.

Event-driven architectures ultimately rely on sensors that are embedded in devices or other types of physical objects. To maximize the effectiveness of the event-driven system, the architecture must be designed such that the sensor data is accessible to all other applications. For event-driven systems to supplant systems based on data-centric architectures, it will be also necessary to combine legacy enterprise data and event-driven data to inform enterprise workflows.

As an IoT Platform, Foundry orchestrates machine and sensor data and enables event-based logic that informs other core systems in the enterprise. Whereas traditional IoT platforms tend to reflect fragmented and immature technologies that require complex integration efforts, Foundry abstracts the complexity of underlying cloud architecture, dramatically simplifies the user experience, and transforms application development from onerous coding to mere clicking of simplified menu options.

Foundry is designed to interact seamlessly with each of the other aforementioned enterprise platform types. In order to support these other platforms, Foundry's services must be both flexible and stable. These attributes allow Foundry to serve as a reliable tool for developing and managing IoT solutions and connecting them to other types of business solutions. By applying serverless architecture concepts and abstracting IoT-centric concepts, Foundry can expand its reach far beyond an IoT-only solution.

Foundry is Inherently Serverless

Gartner actively analyzes the current and emerging serverless ecosystem and has outlined a variety of elements that reflect core attributes and differentiating features of serverless solution architectures. Foundry aligns with these elements, which we believe helps validate our approach. Some highlights that illustrate this alignment are included below.

Event-driven Approach

Digital twins, IoT solutions and conversational AI platforms (such as Microsoft's Cortana, Google Now, Apple's Siri and Amazon Echo/Alexa) require an event-driven approach. Foundry is completely built on top of an event-driven architecture. By focusing on the IoT platform first, our solution is clearly differentiated in the market. We have been able to create a system that is quite different than any other platform on the market.

Gartner writes:

> *Most production systems are designed for web APIs and request-driven synchronous application architectures, including most service-oriented architecture (SOA) implementations and REST-based design. Serverless approaches will shift to an "events first, response second" approach over during the next five years. Both models are essential to modern business, but in the intelligent digital mesh, the main focus will shift toward the event-driven model (see Figure 1). For example, responding in real time to a distress signal from a home device, changing trucking itineraries in response to road conditions or real time route optimization data new road or weather information, orand providing "live" purchase order support. These possibilities can empower customers and create a differentiating business advantage.*

AWS API

Foundry utilizes the AWS API gateway as a means to interact with Foundry from outside systems as well as for Foundry Core functions.

Gartner writes:

> *To adopt a serverless architecture effectively, teams must develop an API that encapsulates a set of microservices that together implement a capability. This miniservice surfaces the capability (to mobile apps, for example), while the microservices implement the individual features within the capability. Applications themselves may expose a higher-level set of APIs that don't expose all the underlying miniservice APIs."*

OS Containers

Gartner writes:

> *Typically, OS containers would be used as an approach to providing a higher level of abstraction above the virtual machine. Serverless computing is a more sophisticated abstraction model building on these concepts. In the serverless cloud computing model, the provider fully manages the infrastructure (for example, virtual machines) to serve application requests so that the developer doesn't have to think about the server resources - even cloud based servers."*

ThingLogix exploits this model to deliver Foundry in a unique way, by combining the strengths of both multi-tenancy and single-tenant application stacks. Foundry deploys into the customer's AWS instance through AWS' CloudFormation service. This allows customers to maintain control over their own data, IoT device certificates, and customizations. This also allows ThingLogix to support a more agile and flexible environment and adapt to rapidly-changing features offered by cloud platform providers.

Foundry Powers Conversational Applications in The Enterprise

Voice-actuated applications are an emerging computing phenomenon that stands to redefine the user experience, stream line workflows, and drive new levels of efficiency in both business and consumer environments. These applications accelerate the emergence of pervasive computing, a longstanding vision that the TV and film series Star Trek first popularized. Pervasive computing is characterized by technology being ubiquitous but essentially invisible to the end user, as it relies on software and computing hardware embedded in objects and voice interfaces instead of tactile interfaces. Machine learning and artificial intelligence technologies also play a critical complementary role in the pervasive computing world, as they will drive accelerated, proactive, and often autonomous actions that quickly eclipse the awareness, accuracy, efficiency, and objectivity of humans. ThingLogix Foundry is strongly positioned to enable pervasive computing for enterprises embracing this trend.

Voice-actuated applications offer a frictionless, simplified user experience based on the system adapting to the user rather than the user needing to adapt to the system. Foundry detects patterns in the user's behavior, asks questions to clarify the user's requests, provides unsolicited and meaningful suggestions, and autonomously takes action on the user's behalf. With Foundry, conversation-enabled applications migrate from reliance on a limited lexicon of fixed commands to more fluid and natural communications between people and technology elements – such as bots, agents, assistants, applications, and other manifestations of technology-enabled services.

Foundry powers voice applications in many ways. It features tools and services that support immersive, continuous, and contextual user experiences through voice services such as those offered by Amazon (Alexa), Google (Assistant), and Microsoft (Cortana). By utilizing natural conversation to interface with enterprise systems, the end user of the Foundry-enabled IoT application need not recall specific commands, terminology, or rules to engage with the system effectively. Additionally, Foundry utilizes AI capabilities that allow the supported application to learn over time from its interactions with both humans and machine-generated data. These capabilities enable continuous learning for the enterprise system and create a virtuous cycle of improved efficiency, user experience, and by extension, business results over time.

Foundry serves as the underlying application platform that facilitates communication and action among users, objects, and applications. It enables the voice application to interface not only with applications, but also with individual objects and systems, as well as their associated data and analytics functions. As the hub for an enterprise’s IoT ecosystem, Foundry provides data from smart, connected devices and objects (input), while driving authorized actions (either predefined or AI-based) that affect the physical world (output). By incorporating voice-actuation capabilities, Foundry enhances the automation and efficiency of the enterprise ecosystems it supports, while also delivering a markedly richer and more enjoyable user experience.

Foundry Is An Event-driven Application Platform

As industry analyst 451 Research noted in a recent report, "Serverless is more than just hype; it has the potential to revolutionize the way we develop, build, and operate applications in the cloud." ThingLogix couldn't agree more. We based our initial vision for Foundry on a 100% cloud-based architecture because we saw that this was simply the most efficient way to abstract the complexity of orchestrating the underlying IaaS layer, simplify and accelerate the development and operation of market-facing IoT solutions at the application layer, prevent costly and time-consuming rework for scaling and expanding an enterprise's IoT solution footprint, and liberate product development and product management teams from technical drudgery so that they could focus on driving business success from IoT.

The serverless phenomenon represents a logical plot point on the evolutionary and revolutionary trajectory that is cloud computing. Without the industry-wide investments in costly on-premise datacenters and enterprise applications, we would never have had the basis for comparing this solution architecture to the alternative of a centralized, often virtualized, often hybridized cloud architecture. Without early adopter businesses willing to risk a perceived lack of control and security in exchange for agility and cost efficiency, we would have never have had a viable market for public cloud services. Without Amazon and Google entering the public cloud services market as an almost accidental extension of their core businesses at the time, we would never have had access to the massive installed base that formerly powered only their respective internal systems.

The current awareness of serverless and the excitement that is steadily building around it reflects an acknowledgement that the cloud is an established alternative to on-premise architectures that also offers its own vast and unique advantages. It is also an acknowledgement that there is still room for improvement and optimization in cloud architectures. Serverless architectures complement other cloud-centric trends, such as microservices, containers, cloud-native DevOps and DevSecOps, and the broad topic of digital transformation in the enterprise.

ThingLogix chose to optimize the serverless environment of AWS due to three principal factors. First, we are historically related to the company that AWS acquired and subsequently positioned as the cornerstone of AWS IoT. Second, we observed firsthand the complexity of orchestrating the myriad services that AWS offers – both IoT-centric and other IoT-relevant services – for both development and ongoing management of IoT applications. We thought this was a challenge worth solving. Third, AWS has steadily and unquestionably emerged as the market leader in public cloud infrastructure services globally, and we feel confident that it makes sense for us to focus where the customers are.

But serverless is bigger than any one company, be it AWS or ThingLogix. We have architected Foundry to be a world-class, enterprise-grade, cloud-based solution development and management platform for IoT. We're not the only company in our market, but we think our approach is both unique and superior. As the IoT market continues to grow, and as enterprises accelerate their adoption of serverless architectures, we believe that we will occupy a commanding seat at the table for enterprises seeking to create more value from their IoT solutions and other enterprise platforms.

Gartner

In their 2017 Hype Cycle report, Gartner identifies IoT Platforms as being at the peak and Serverless PaaS as being on the rise. Thinglogix believes that these two trends will converge and that the serverless PaaS will be the foundation for stability and pervasive adoption of IoT in the Enterprise.

Select quotes:

- *The key trends in PaaS that are driven by increasing investment in IoT and digital business include: To support IoT requirements, vendors are expanding their cloud services to include not just the IoT device gateways, but also event messaging, data quality, data integration, decision rule engines, stream analytics, predictive and prescriptive analytics, and in-memory data management. This accelerates the emergence of the comprehensive PaaS suites, integrating the previously separately standing specialized xPaaSs (see "Predicts 2017: IoT Proliferation Will Drive Investment in New Approaches to Implementation").*
- *The digital twin, a new architecture model, is becoming essential to IoT solutions and platforms, and most vendors are actively developing effective digital twin architecture and services. Digital twin promises to become a pervasive part of cloud platform services (see "12 Core Principles of Application Architecture for Digital Business and IoT").*
- *Pure cloud vendors are increasingly under pressure to support the hybrid cloud/local platform model. To support IoT business, vendors are being required to offer software for local deployment at the IoT edge (see "Use the IoT Platform Reference Model to Plan Your IoT Business Solutions"). In response, cloud vendors are adapting: Amazon Web Services (AWS), previously a pure cloud vendor, now for the first time offers a software product (AWS Greengrass), and Google has become a software vendor with its acquisition of API management vendor Apigee.*
- *Event processing is becoming recognized as a new, required capability for any cloud platform provider aspiring to market leadership. IoT can enhance organizations' real-time situation awareness, but to exploit the information streams emanating from IoT devices, platforms must be equipped with event and event stream processing capabilities. Apache Kafka, Apache Spark, Apache Storm and other event-related technologies, including the function platforms (such as Iron.io or AWS Lambda), are fast-becoming ubiquitous in cloud platform portfolios. [See "Top 10 Strategic Technology Trends for 2017"]*
- *Encouraged by breakthroughs in consumer IT, organizations are seeking greater responsiveness and intelligence for their automated or automation-assisted decisions, many of which are triggered by IoT events. xPaaSs, which are focused on business analytics (business analytics PaaS [baPaaS] such as ClearStory Data), are some of the fastest-growing types of cloud platform services. They are supported by hundreds of data broker services (DaaS), such as Acxiom, and by data broker PaaSs, such as Flybits, and AI accelerator services such as IBM Watson. [See "The Platform Architect's Guide to Designing IoT Solutions"]*
- *The massive amount of data generated by some IoT solutions often ends up in cloud-based data stores (in database PaaS [dbPaaSs], such as Amazon DynamoDB or Google's Cloud Bigtable), thus increasing the adoption of cloud data stores and improving the trust that mainstream organizations have in cloud data management. Data integrity and security are often stated as the barrier to general cloud adoption, so the increasing use of dbPaaSs contributes to the increasing acceptance of cloud platforms overall in mainstream organizations. [See "Internet of Things — Architecture Remains a Core Opportunity and Challenge: A Gartner Trend Insight Report"]*

- *Building a vibrant SaaS ISV ecosystem is a prerequisite for cloud platform market dominance. Yet, of the major cloud platform providers, only Salesforce has a significant ecosystem of partner SaaS offerings built on its cloud platform (Salesforce App Cloud). This differentiation helps Salesforce maintain its PaaS market leadership, while the lack of a comparable ecosystem holds the others back. [See "The Key Trends in PaaS, 2017"]*[1]
- *The cloud megavendors, Salesforce and SAP, have no general IaaS offering, and AWS has no SaaS or organized SaaS partner ecosystem (although many ISV customers host their applications on AWS EC2). Yet, a full spectrum of cloud platform capability, including IaaS, PaaS and SaaS components, is a likely characteristic of a long-term cloud platform leader. As a result, vendors such as SAP and Salesforce are investing in strategic partnerships and promoting the virtues of the multicloud model (Salesforce is developing an IaaS partnership with AWS, and SAP targets AWS, Microsoft and Huawei IaaS). [See "The Key Trends in PaaS, 2017"]*
- *AWS has demonstrated the ability to lead with cloud platform innovation, yet for a mainstream enterprise to embrace AWS cloud platform beyond the basic IaaS, it has to leave behind much of its existing practices, investments and skills, and accept... lock-in with an enterprise-unproven vendor — conditions that are easier met for new innovative solutions than for the core enterprise systems. [See "The Key Trends in PaaS, 2017"]*[2]
- *The push toward Microservices enable developers to reduce the scope of an application to an individual feature or function optimized for agility and scalability at a detailed feature level. [See "Top 10 Strategic Technology Trends for 2017"]*

Other Sources

Jamii Forums
Technology providers are already starting to experiment with the symbiotic relationship between the CAP and IoT-related platform services. In its August 2016 update, Microsoft's Skype division added the "If This Then That" (IFTTT) bot to its bot directory. It can interact with more than 50 different types of IoT devices, ranging from cars to wearables to connected home devices.

Pivotal

What is an event-driven architecture?

"Events" typically refer to things that happen. For instance, a sensor reports a temperature change, a user clicks their mouse, a customer deposits a check into a bank account. In an event-driven architecture, systems emit and process events in a loosely-coupled fashion. These environments promote asynchronous communication and adaptability.

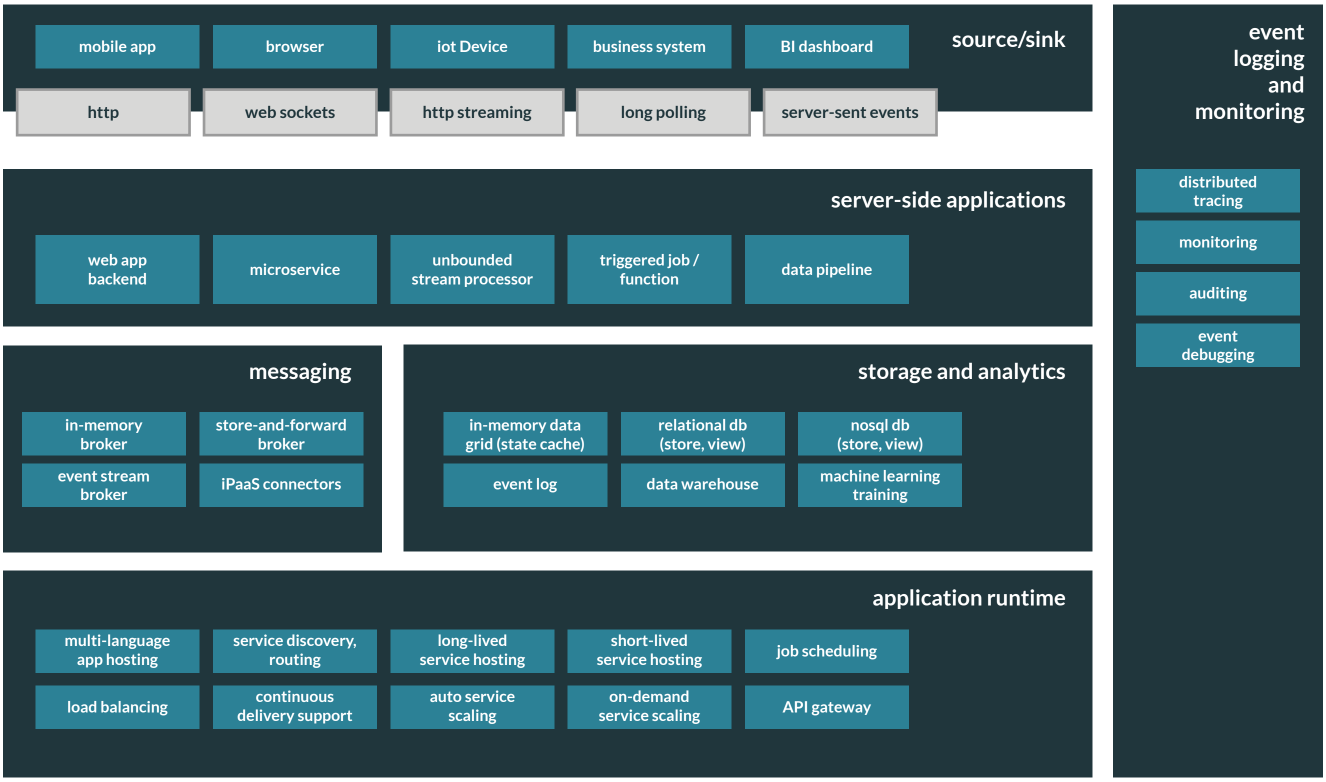
An event-driven architecture transforms how you generate, process, store, and access data. While possibly more complex to understand, an event-driven architecture helps you embrace unpredictability and reduces dependencies between components. As a result, you can adjust to business changes faster. Martin Fowler of ThoughtWorks does an expert job explaining these core principles in a recent blog post and follow-up video.

So in this model, events are core to your architecture. Being "event-driven" means *much* more than installing Apache Kafka or using serverless Functions. It involves a catalog of capabilities that together help you emit and process events. Here's one visualization of those capabilities.

[1] *From ThingLogix: The architects behind Foundry have deeply-rooted experience in the Salesforce ecosystem. and in many ways have replicated the Salesforce value proposition in the IoT and serverless context.*

[2] *From ThingLogix: The enterprise must accept a commitment to a new cloud-based architectural model that requires significant changes to legacy systems. These conditions are more achievable during the pursuit of new and innovative solutions, not when replacing or enhancing legacy systems.*

Event-Driven Reference Architecture



source/sink

mobile app · browser · iot Device · business system · BI dashboard

http · web sockets · http streaming · long polling · server-sent events

server-side applications

web app backend · microservice · unbounded stream processor · triggered job / function · data pipeline

messaging

in-memory broker · store-and-forward broker · event stream broker · iPaaS connectors

storage and analytics

in-memory data grid (state cache) · relational db (store, view) · nosql db (store, view) · event log · data warehouse · machine learning training

application runtime

multi-language app hosting · service discovery, routing · long-lived service hosting · short-lived service hosting · job scheduling · load balancing · continuous delivery support · auto service scaling · on-demand service scaling · API gateway

event logging and monitoring

distributed tracing · monitoring · auditing · event debugging

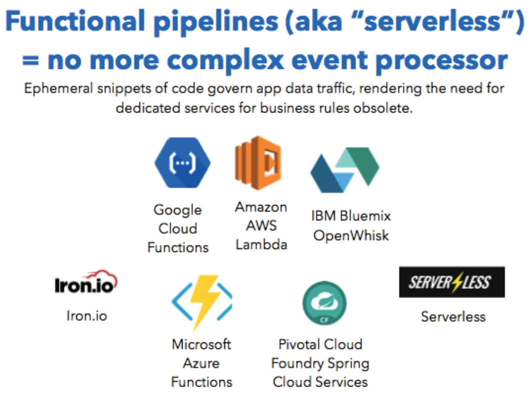
Research and Markets

The Function-as-a-Service (FaaS) market size is estimated to grow from USD 1.88 billion in 2016 to USD 7.72 billion by 2021, at a Compound Annual Growth Rate (CAGR) of 32.7% and IoT is forecast to grow to $267B by 2020. Thinglogix is positioned to take advantage of the advancement of two major computing trends feeding these market projections and identified by Gartner as described in the 2017 Hype Cycle assessment:

- IoT Platforms
- Serverless (Event Driven) PaaS

Work-Bench

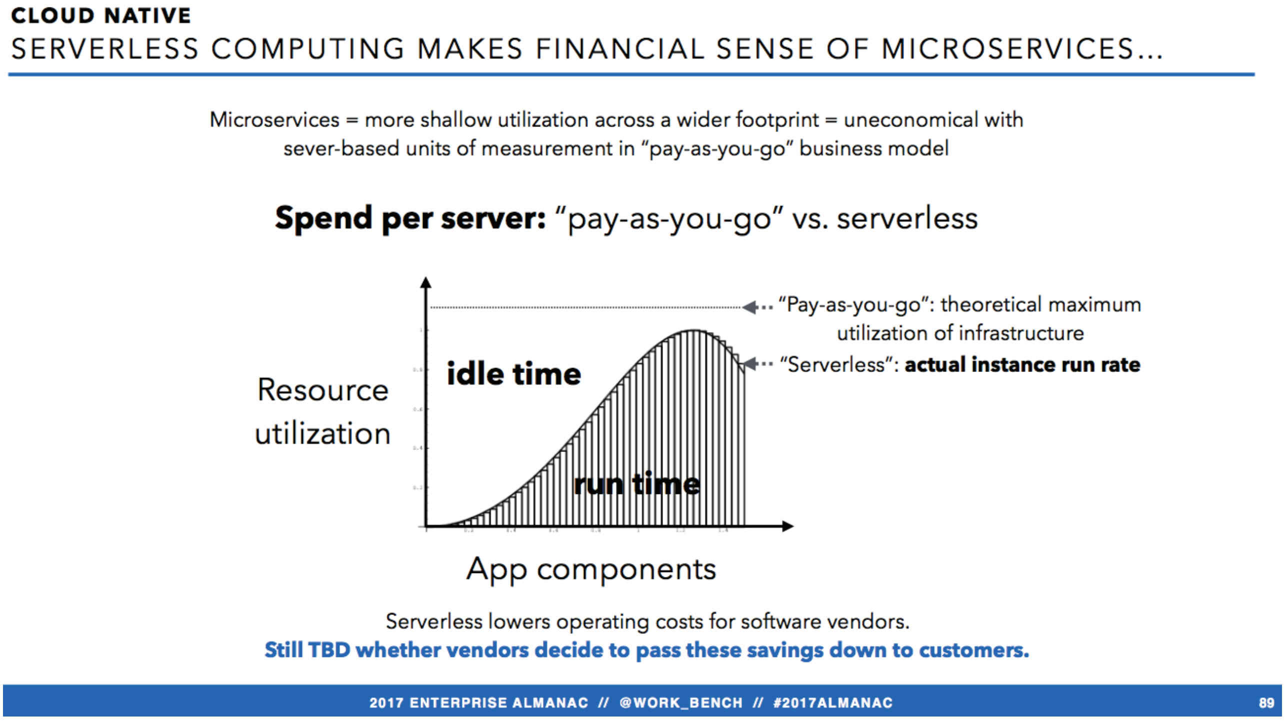
Middleware, once a core layer of the IT stack, is shedding significant weight as middleware functions now reside in distributed code.



Functional pipelines (aka "serverless") = no more complex event processor

Ephemeral snippets of code govern app data traffic, rendering the need for dedicated services for business rules obsolete.

Google Cloud Functions
Amazon AWS Lambda
IBM Bluemix OpenWhisk
Iron.io
Iron.io
Microsoft Azure Functions
Pivotal Cloud Foundry Spring Cloud Services
SERVERLESS
Serverless



CLOUD NATIVE

SERVERLESS COMPUTING MAKES FINANCIAL SENSE OF MICROSERVICES…

Microservices = more shallow utilization across a wider footprint = uneconomical with sever-based units of measurement in "pay-as-you-go" business model

Spend per server: "pay-as-you-go" vs. serverless

Serverless lowers operating costs for software vendors.

Still TBD whether vendors decide to pass these savings down to customers.

2017 ENTERPRISE ALMANAC // @WORK_BENCH // #2017ALMANAC 89