Above the Clouds

Re-platforming for a new era of unified, automated, and agile business operations

David S. Linthicum

Sponsored by

EnterpriseWeb®

Hitting the Cloud Complexity Wall

Two decades ago, Amazon launched the cloud in 2002. AWS was the first to offer remote virtualized pools of computing resources exposed as web services on a utility basis. The initial response from the IT industry was lukewarm, if not outright skeptical.

It took a decade for the cloud to become mainstream. By 2012, more development teams and users were taking advantage of quick access to cloud-based infrastructure and low initial cloud costs to accelerate service delivery, often using corporate credit cards to bypass their internal Ops teams. The lure of convenience drove the decentralization of the IT estate.

No one disputes that the cloud and cloud-native design principles are a boon to AppDev productivity and responsiveness, though not without side effects. Distributing the IT estate and disaggregating applications led straight to today's complexity challenges around multicloud (aka, hybrid cloud).

The average of 500-800 services under operational management within a single cloud deployment rose to 2000-3200 with multicloud deployments. Consequently, IT must deal with 3-4 times the operational complexity.

The hyperscalers don't make multicloud easy. Each cloud is a walled garden with its own native interfaces. The more a solution integrates into a particular hyperscaler's services, the less portable that application becomes. Lock-in concerns aside, hyperscalers create islands of automation that inhibit interoperability and governance despite the containers' promise of Kubernetes.

Contrary to the hype, not everything runs in the cloud or on containers. The modern enterprise must cope with a highly distributed, diverse, and dynamic computing environment that includes:

- Clouds, data centers, and edge deployments
- Server, VM, and container-based applications
- SDN, VPC, SD-WAN, and service mesh networks
- Internal, partner, cloud, and web services

Collectively, these technical silos impede the use of value streams. This considering that value streams are really just a series of steps that an organization uses to implement solutions that provide a continuous flow of value to a customer.

Organizations struggle to reconnect their fragmented operations, automate end-to-end processes, and establish consistent policy management at scale. They've hit a complexity wall and desperately need new solutions to flexibly connect and continuously optimize their value streams.

Getting on top of the problem

The rise of multicloud revealed and compounded an existing distributed systems problem. IT has been reintegrating and networking what it just distributed since the enterprise began to leave the mainframe. The business gained some modularity and choice, but at a cost. As developers disaggregated monoliths into services, then into microservices, and then serverless functions, the number of elements to connect and manage exploded. We can no longer get away with manually and statically coding components back into systems. IT cannot keep up.

Enterprise IT needs a layer above clouds, data centers, and edge deployments for an enterprise-wide view of its distributed systems. A simple registry won't cut it. Without a higher level of abstraction and a shared model across teams, environments, and applications, developers are still left with point-to-point integration, manual debugging, and management.

Developers need powerful and practical abstractions that hide complexity and streamline their work. Today's trends point to a common services layer that's often referred to as a "Metacloud" (aka, "Supercloud"). Platforms will arise to bring order to this space with logical models and support for cross-cloud automation, management, and governance.

The metacloud is a cross-cloud service layer that eliminates the need to define hyperscaler-specific technologies for each public cloud deployment. A metacloud will reduce complexity and eliminates redundant tasks such as working with specific native security technology within each cloud provider. It uses common abstraction and automation layers to deal with operations, governance, and security with a single pane of glass and a single set of APIs, so these systems are much easier to manage. Some benefits of a metacloud include:

- Common operations services (gateway, broker, transformation, configuration, and workflow automation) that carry out operations for each cloud provider via a unified interface and consistent developer experience.
- General operations, such as application, service, infrastructure, and network management.
- Cross-cloud orchestration, including container orchestration and management.
- DevOps services.
- Support for specific types of operations such as security, monitoring & observability, AI/ML, etc.
- Data services that integrate, orchestrate, and manage data.
- A common schema and shared memory that spans clouds for interoperability, analytics, and optimizations.

Most enterprises want this logical layer of abstraction, orchestration, and automation that runs above all cloud providers (see Figure 1). Their ideal metacloud would include legacy systems as well as edge-based systems.

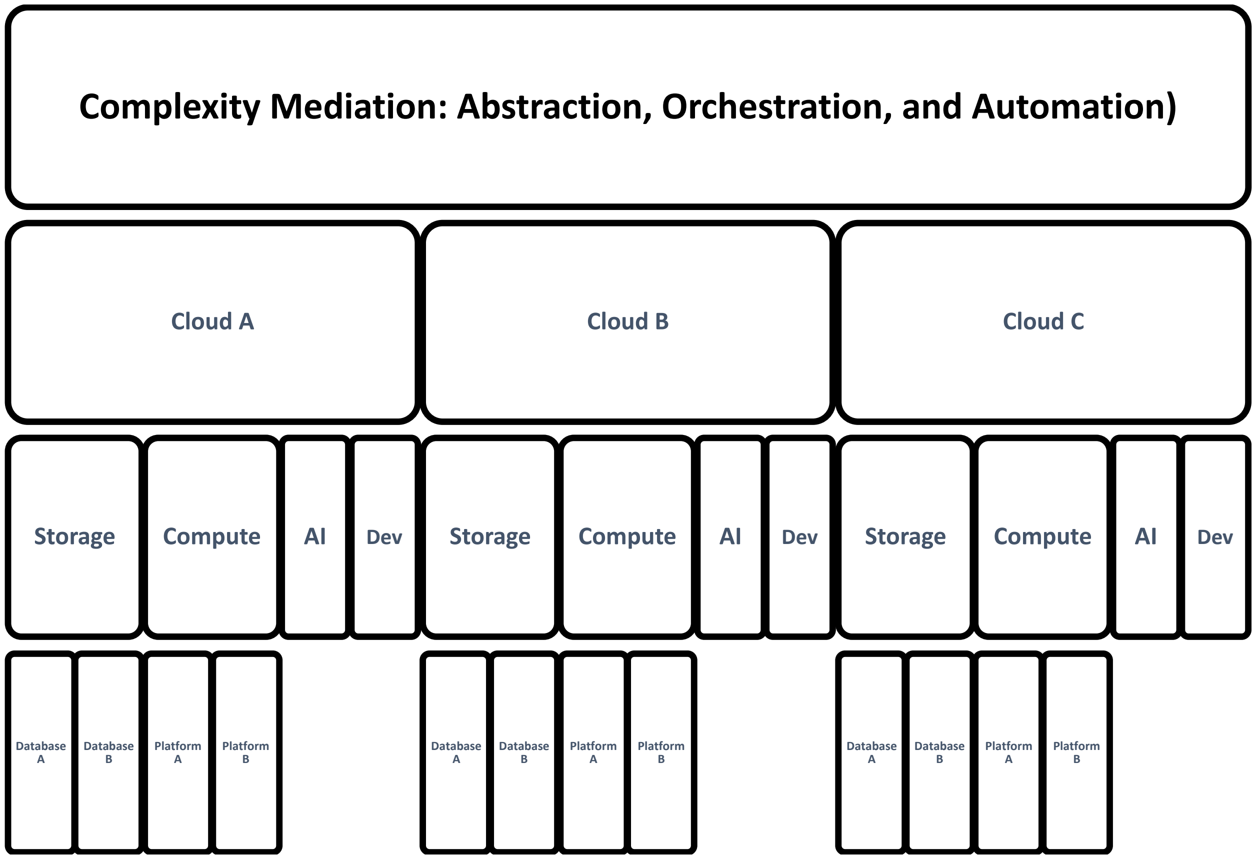

Figure 1: Multiclouds require an abstraction, orchestration, and automation layer to mediate the complexity of siloed cloud services. This includes redundant data, services, security, operations, and governance services, along with other resources and services that need to be managed using a single unified layer: A metacloud.

As of this paper's publication date, the metacloud layer is being recognized as a necessity to control the complexity of distributed and heterogenous architectures, including multicloud architectures. The core benefit is to provide cross-cloud services that eliminate redundancy and lock-in, and abstract native APIs so developers can design infrastructure-independent and protocol-agnostic services that help realize the cloud-native vision. The name for this layer might change, but the core principles and values will remain the same.

Finding Common Ground

The fundamental challenge for a metacloud platform is to enable consistent discovery, composition, and management of heterogeneous elements. To steal a line from the movie Cool Hand Luke – "What we have here is a failure to communicate."

The communications problem requires a form of universal translator, a common intermediary language that will mediate communications with and between elements to facilitate interoperability, end-to-end automation, and common lifecycle management.

A consistency layer requires a common language to represent heterogeneous solution elements as typed objects described by shared metadata and relationships. A single source of truth reduces developers' cognitive overload by abstracting away differences, leaving it to the runtime to handle the implementation details.

A human- and machine-readable common language eliminates tedious integration, transformation, and configuration tasks. It supports a design environment for developers to declaratively compose objects from a catalog into infrastructure-independent and protocol-agnostic intent-based services. It also allows an execution environment to use the same metadata, relationships, and types to translate policies and drive automation. This opens the door to a platform that offers common services across host environments, abstracting their native interfaces. The runtime handles the deployment, configuration, and management based on the developer's intent (i.e., modeled service topology along with service chain, and SLA and LCM policies).

The Devil is in the Details

Of course, building a universal translator is easier said than done. The scope is not just end-to-end across high-level cloud-native interfaces, it's also top-to-bottom to automate connections, and to automate integration and configuration at the application (or service) layers, infrastructure layers, and network layers. In other words, you can't use only horizontal lines to draw a plaid pattern. A partial solution will leave low-level details unmodeled and unautomated. The lack of a common language explains why today's prevailing approach is still Infrastructure-as-Code.

Clouds and containers are only part of the enterprise IT mix. Most organizations have sprawling and diverse IT estate which includes mainframes, servers, virtual machines, and container-based processes.

Kubernetes itself does not fit all workloads. Even when it is adopted, a lot happens above the clusters it manages that must be addressed by Ops teams and, to some extent, developers themselves. With advances in programmable networking, developers now have API access to the hardware itself below the cluster, which allows application-driven optimizations that could radically improve performance and reduce energy consumption. However, this powerful feature remains impractical without high-level abstractions that support automated policy-based dynamic configurations of the infrastructure (*e.g.,* NICs, processors).

Declarative composability requires an abstraction over this whole system's domain end-to-end and top-to-bottom so that any arbitrary composition of heterogenous objects can be understood by the system, and then deployed and managed in diverse environments. Gaps in the model would require human intervention, which introduces tight coupling and undermines the rationale for the metacloud layer.

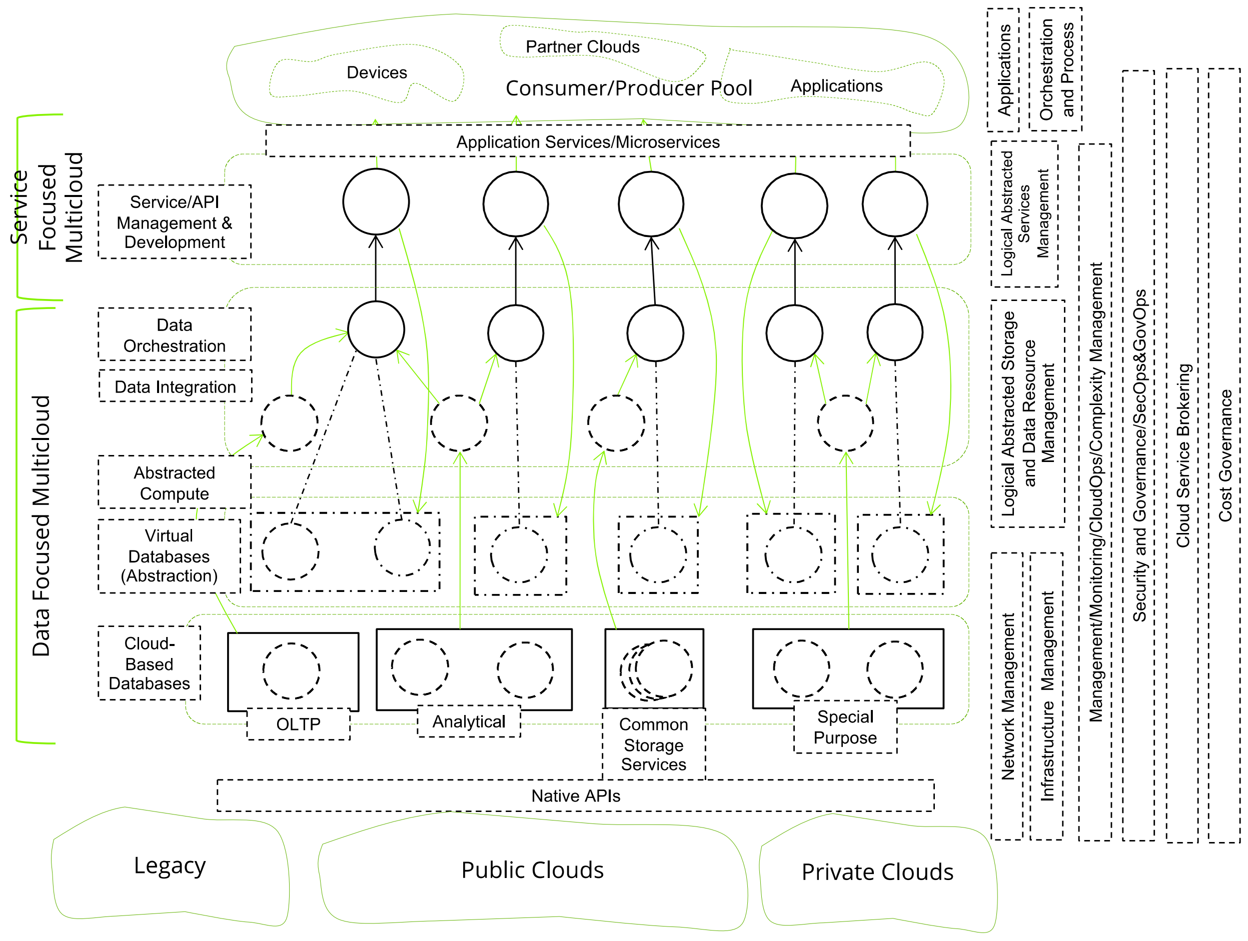
Partner Clouds
Devices
Consumer/Producer Pool
Applications
Application Services/Microservices
Service Focused Multicloud
Service/API Management & Development
Data Focused Multicloud
Data Orchestration
Data Integration
Abstracted Compute
Virtual Databases (Abstraction)
Cloud-Based Databases
OLTP
Analytical
Common Storage Services
Special Purpose
Native APIs
Legacy
Public Clouds
Private Clouds
Applications
Orchestration and Process
Logical Abstracted Services Management
Logical Abstracted Storage and Data Resource Management
Network Management
Infrastructure Management
Management/Monitoring/CloudOps/Complexity Management
Security and Governance/SecOps&GovOps
Cloud Service Brokering
Cost Governance

Figure 2: Metaclouds abstract application, data, and infrastructure complexity so developers can focus on the business

Metacloud at the Crossroads

A metacloud is a high-level control plane over-the-top of clouds, data centers, and edge sites. However, centralized top-down command and control systems are not the only features that drive metacloud development. A metacloud's abstractions and common services eliminate tedious work to make self-service systems that accelerate service delivery and facilitate interoperability and governance. Metacloud is a force multiplier that must draw on related IT movements such as Data Mesh, SRE (Site Reliability Engineering), and DevSecOps to help bring them together.

SRE focuses on infrastructure reliability as points in the solution rather than holistically focusing on the solution and the infrastructure. This is a shift in thinking about how we carry out SRE. It will work more in and between points of infrastructure which will provide the best value for the money and effort spent on SRE. Metacloud can be both a source and target for SRE optimizations.

Data mesh is a sociotechnical approach to building decentralized data architectures. Much like SRE, the data mesh focus has been on the endpoints, being a domain-oriented and self-service design. With the focus on points in the middle of the solution, in this case, data, the use of data mesh provides more added value because it focuses on the common features of the data rather than the physical instances. The benefit of this evolved approach is its promotion of data "products," common services, and developer self-service. A metacloud incorporates and expands on these values as a catalog-driven platform for applications.

Perhaps most importantly, metacloud is the natural intersection of DevSecOps automation and service, infrastructure, and network management. Here an end-to-end and top-to-bottom abstraction provides common metadata for Day 0 Design, Day 1 Deployment, and Day 2 Operations. It eliminates silos, streamlines handshakes between teams, and improves troubleshooting. This type of abstraction also supports automated integration and configuration at all levels and connection automation.

Of course, given the scope and scale of distributed systems, a seamlessly functioning metacloud platform is a big ask. If it were simple, it would have been done long ago. These are early days and solutions are just starting to emerge. The good news? There is one company that's been working on complex distributed systems for a decade. EnterpriseWeb has a rich and mature platform product that is well-positioned to take on the metacloud challenge.

A Platform for Transformation

EnterpriseWeb® is an interesting software company that we've known about since before "The Cloud" was cool. Their CEO, Dave Duggal, saw the increasing fragmentation of business operations and he anticipated the growth of distributed systems challenges. He founded the company with a mission to abstract complexity and enable logically centralized discovery, coordination, and management.

I've watched countless startups pivot from their mission statements to chase increased sales and/or markets, only to implode. EnterpriseWeb stubbornly stuck to its developer platform vision, executed against it, and waited for the world to catch up. Now EnterpriseWeb finds itself in the right place at the right time. They validated their technology in one of the most demanding markets (telecom), and have impressive partners that include Intel®, Red Hat®, and Tech Mahindra®.

The telecom industry has the same multicloud challenges those other industries experience, but telecom headaches are compounded by the scale, complexity, and performance expectations of network infrastructure. The solutions for highly automated and agile telco operations are likewise rooted in high-level abstraction and advanced cloud-native connectivity, coordination, and configuration capabilities.

EnterpriseWeb offers an industrial-grade no-code platform for software engineers.

The company has pioneered the use of graphs to model behavior and data which in turn supports agile integration and automation across applications, services, cloud hosts, networks, and physical devices.

The platform is based on the company's proprietary language, GOAL™ (Graph Object and Action Language). The graph provides shared human and machine-readable metadata and relationships to both a declarative design environment as well as a cloud-native execution environment. GOAL is a dynamically interpreted language so there is no build or compile.

Conceptually, EnterpriseWeb is a Graph DSL that supports an intelligent IDE that has an integrated execution environment, all in one 50Mb package.

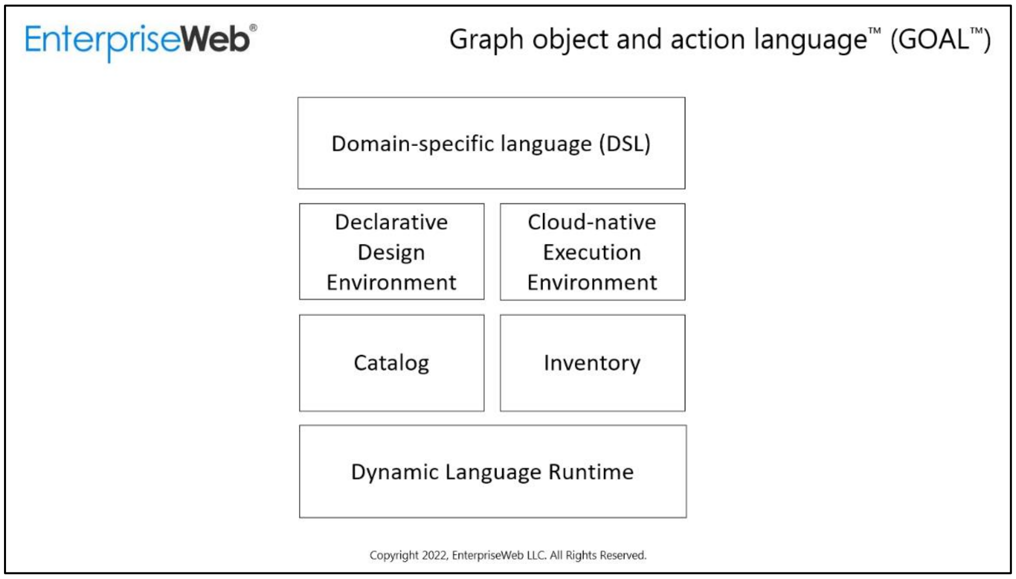
EnterpriseWeb®
Graph object and action language™ (GOAL™)
Domain-specific language (DSL)
Declarative Design Environment
Cloud-native Execution Environment
Catalog
Inventory
Dynamic Language Runtime
Copyright 2022, EnterpriseWeb LLC. All Rights Reserved.

Figure 3: A dynamic software language provides the platform's underlying ability to model and run complex distributed systems

The cloud-native platform deploys as a cluster of pods. It's RHEL8- and operator-certified by Red Hat and runs on Red Hat's lightweight Java framework, Quarkus. The platform is API-first via a REST client and can run headless behind customer portals, but it also has rich design and management consoles.

The platform provides a ready-to-use graph knowledge base with generic business concepts and system types to jump-start customer solutions. The out-of-the-box model provides the end-to-end and top-to-bottom metadata and relationships necessary to fully automate DevSecOps. Solution architects can import their existing information models and manually curate the graph knowledge base to extend the domain for their specific industries and operations.

This umbrella abstraction (an enterprise 'web') provides a unified interface to weave distributed data and functions into a logical view with aggregated system-wide entities, and an indexed catalog of typed domain objects.

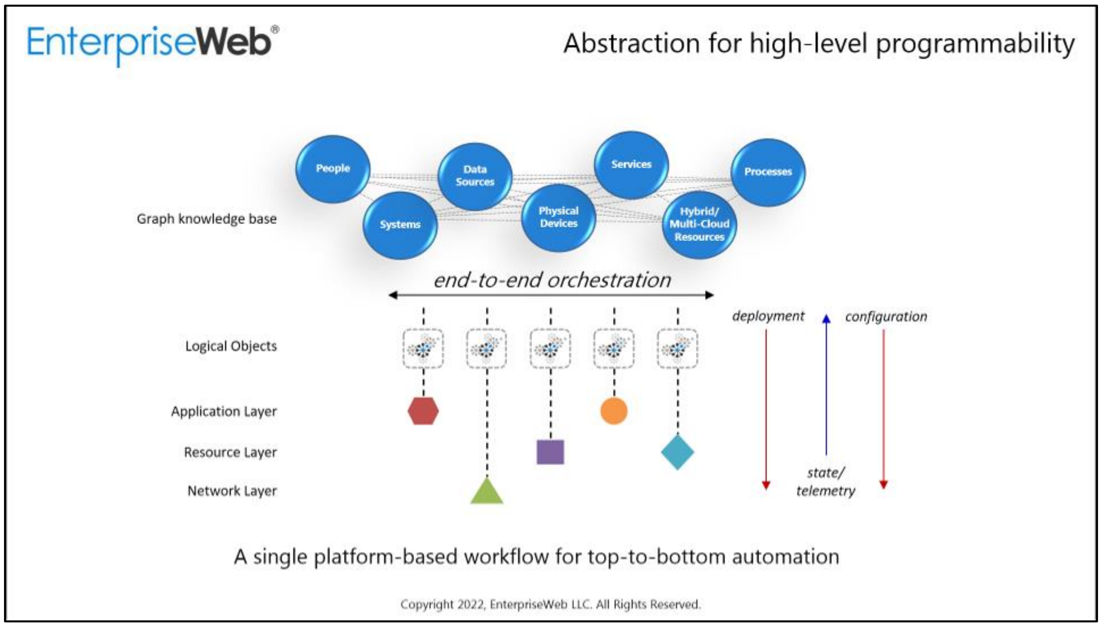

Figure 4: An umbrella abstraction enables global transparency, end-to-end coordination and policy-based management

The design environment allows customers to rapidly model solution elements as typed objects registered in a catalog and declaratively compose objects into intent-based services that are infrastructure-independent and protocol-agnostic. The platform generates operators for all the participating elements and the platform dynamically configures the CRDs when a host is attached to the service.

All objects are wrapped in software contracts with pre- and post-conditions that support system-level policies.

- Security/Identity (IAM, Certs/Auths, Secrets)
- Reliable messaging (idempotent, immutable)
- Transaction guarantees (retries, compensations, trace)
- State management (log-style, append-only persistence, aggregates, durable entities, dynamically indexed and tagged, with time series and audit history)

The execution environment also offers a collection of common services through a novel "Middleware-as-a-Service" approach.

EnterpriseWeb's platform services represent a comprehensive set of MoM (Message-oriented Middleware) patterns implemented as a library of event-driven, stateless functions (FaaS) that are exposed by REST APIs. This is an interesting approach – lightweight, stateless, and scalable.

The platform's type system attaches middleware behavior when it responds to requests and events. The execution environment dynamically dispatches and coordinates the functions as a dataflow process. The graph-connected knowledge base acts as shared memory to efficiently hydrate session state and global context to configure the functions for specific executions.

This approach supports highly dynamic interactions where the middleware capabilities self-organize based on context and then go away. There are no long-running threads, no stacks, no hops, and no context-switching. A complex task could include hundreds of parallel operations in milliseconds to instantiate, connect, integrate, configure, and manage participating solution elements at the application, infrastructure, and network layer.

As an integration and automation platform, EnterpriseWeb is open by design. The platform supports delegation to federated 3rd-party components, legacy systems, and physical functions/devices.

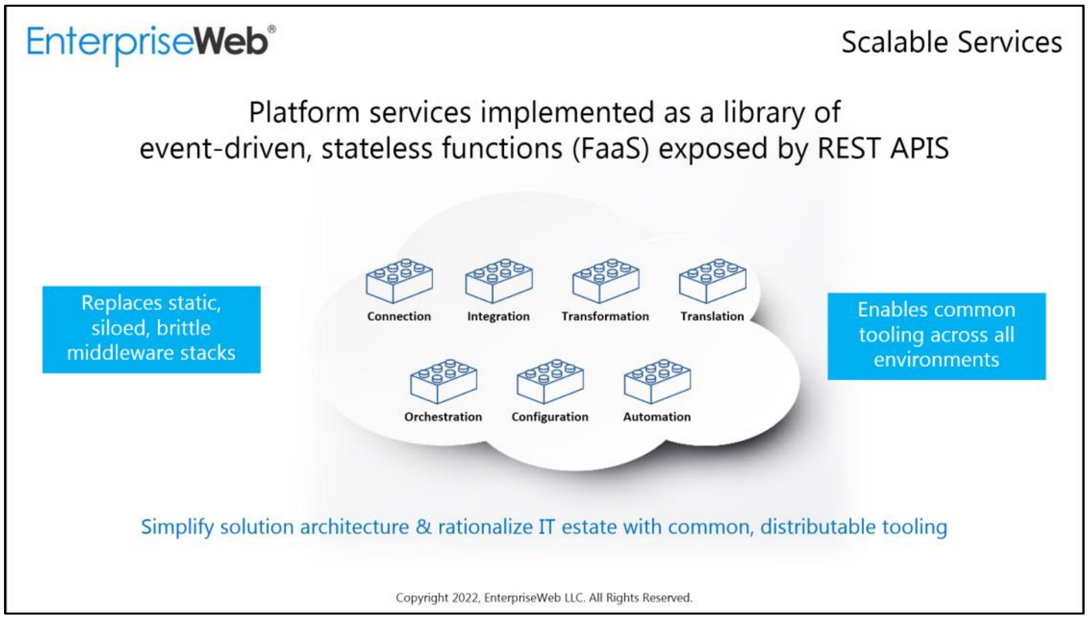
EnterpriseWeb®
Scalable Services
Platform services implemented as a library of event-driven, stateless functions (FaaS) exposed by REST APIS
Replaces static, siloed, brittle middleware stacks
Connection
Integration
Transformation
Translation
Orchestration
Configuration
Automation
Enables common tooling across all environments
Simplify solution architecture & rationalize IT estate with common, distributable tooling
Copyright 2022, EnterpriseWeb LLC. All Rights Reserved.

Figure 5: The platform leverages the graph knowledge base to efficiently hydrate context for stateless functions

Middleware-as-a-Service presents an opportunity to have common tooling and consistent developer experiences across all environments. It provides lightweight, low-latency, high-performance, distributable middleware for the cloud, data center, and edge sites. As a serverless framework, the platform scales down to near zero when not in use.

EnterpriseWeb nodes form a fabric akin to a service mesh, but for integration and automation "middleware" services. Middleware-as-a-Service greatly simplifies solution architecture, accelerates deployments, and rationalizes IT estate.

EnterpriseWeb avoids the need to tightly integrate with each cloud provider's services, which is expensive, leads to silos, creates vendor lock-in, and inhibits multicloud initiatives. Instead, EnterpriseWeb customers can bring their own middleware to all their environments!

Conclusion

We have big problems that require sophisticated solutions. It’s hard to make inherently complex distributed systems’ problems easy.

Most companies simply don’t have the time, budget, or expertise to build their own metacloud. Some may gravitate to frameworks, but every CTO knows a framework is not a solution. Metacloud doesn’t lend itself to DIY. In my view, this market is ripe for platforms that enable organizations to start small, quickly demonstrate use cases, and then scale out based on success.

No matter where your enterprise is on its journey to systems modernization and net-new development, many IT departments are trying to dig out of a bottomless hole of complexity. My advice is to stop digging. You can solve IT complexity problems by leveraging resources in a more extensible and scalable manner (i.e., applications, data, processes, services, etc.).

Placing volatility into a domain that evolves through metadata-based configurations is an idea as old as interconnected technology. The ability to execute the idea remained unproven…until now.

Today we can use middleware to leverage abstraction, orchestration, and automation as core services from a centralized control plane that spans all systems, applications, databases, and other assets. This is the most viable means to pull off the expansion of services, applications, and data, and still maintain an operational state. Most importantly, this approach facilitates change and expansion; and leverages best-of-breed services to support business innovation.

It's time to address today’s problems with today’s answers. Take a deeper look at EnterpriseWeb to learn how Middleware-as-a-Service could help your enterprise.

About the author

David Linthicum is on most top 10 lists of technology innovators and influencers, including cloud computing, edge computing, and security concepts. David is a best-selling author of over 15 books and over 7,000 published articles. He is also the originator of many business-related technology concepts, including Enterprise Application Integration (EAI). He's an innovator within Service-Oriented Architecture (SOA), and now cloud computing and the use of cloud computing for digital transformations.

David's 50+ courses on LinkedIn Learning consistently appear on the "Popular Courses" list and provide course content on cloud computing, cloud architecture, cloud security, cloud governance, cloud operations, DevOps, and many other concepts related to cloud computing and enterprise technology in general. He's also an adjunct professor for Louisiana State University (LSU), where he's created courses on DevOps, Cloud Computing, Cloud Architecture, and other courses that are in-demand by the LSU student body. David has done over 1,000 conference presentations in the U.S. and abroad, often as a keynote speaker at conferences related to enterprise technology. He has hosted over 2,000 Webinars on the correct use of enterprise technology, including cloud computing, edge computing, DevOps, and data science.

David's holistic view of the technology value model cuts through the hype cycles and other distractions to provide clear solutions. His goal is to provide opportunities for businesses to lead their markets by weaponizing technology to provide better products, services, and customer experiences. His mission is to use innovative technology approaches that provide the best outcomes for his clients.