

For: Application  
Development  
& Delivery  
Professionals

# Mobile Needs A Four-Tier Engagement Platform

by Michael Facemire, Ted Schadler, and John C. McCarthy, October 18, 2013

## KEY TAKEAWAYS

### **Mobile's Continuous Delivery Requires A Loosely Coupled Architecture**

Embracing mobile opportunities means embracing new set of requirements. Firms must deploy and continuously update their services, adapt to rapid change, and scale up to deliver great performance. Client devices consuming these services must dynamically compose these capabilities in near real time to create an outstanding user experience.

### **Firms Need An Engagement Platform**

The three-tier web architecture won't deliver any more: Mobile experiences require a four-tier engagement architecture. A client element focuses on unique device capabilities; a delivery tier covers the last wireless mile; an aggregation layer integrates services; and a service tier encapsulates local and remote services.

### **Addressing Mobile's New Requirements Also Requires A New Mindset**

Application developers and enterprise architects, and the vendors that serve them, must relentlessly adhere to loosely coupled services and tiers, a distributed deployment and development model, and dynamically composed services to support contextual experiences.

### **Winners And Losers Will Emerge Over The Course Of The Next Five To Six Years**

The last time the industry changed architectures -- from client/server to cloud -- it took 10 years and lots of mistakes. New cloud architectures will cut that migration time by 40% to 50%. Established players and startups will wrestle for relevance as they begin to define and deliver new four-tier engagement architectures.

# Mobile Needs A Four-Tier Engagement Platform

Web Architectures Can't Handle The New Demands Of Engagement

by [Michael Facemire](#), [Ted Schadler](#), and [John C. McCarthy](#)  
with [Jeffrey S. Hammond](#), [James Staten](#), and Rowan Curran

## WHY READ THIS REPORT

Mobile is pushing aging web architectures to the brink. The three-tier architecture built for a browser-led PC world can't flex, scale, or respond to the needs of a good mobile experience or the emerging requirements for connected products. Mobile's volatility and velocity of change require a distributed four-tier architecture that we call an "engagement platform." The engagement platform separates technical capabilities into four parts: client, delivery, aggregation, and services. The new requirements of modern apps will force content distribution networks, application server vendors, mobile middleware vendors, platform-as-a-service suppliers, a myriad of startups, and enterprises to coalesce around this four-tier architecture. CIOs need to start planning immediately for the migration from three tiers to four.

## Table Of Contents

### 2 Mobile Requirements Leave Aging Web Architectures Out In The Cold

Mobile Engagement Demands A New Set Of Capabilities

### 4 Mobile Needs A Four-Tier Engagement Platform And Approach

Engagement Platforms Embrace New Design Principles

### 6 Early Examples Of The Four-Tier Engagement Platform

### 8 How The Engagement Platform Market Will Unfold

Three Vendor Camps Will Vie For The Aggregation Tier Sweet Spot

## RECOMMENDATIONS

### 10 Take Four Steps Toward The Four-Tier Engagement Architecture

## WHAT IT MEANS

### 11 Vendors Will Coalesce Around The Tier They Can Own

### 12 Supplemental Material

## Notes & Resources

Forrester interviewed 90 vendor and user companies, including Akamai Technologies, Antenna Software, AnyPresence, Atos, Cynergy Systems, FeedHenry, IBM, Infor, July Systems, Kinvey, Leapfactor, Microsoft, New Relic, salesforce.com, SapientNitro, StackMob, and Tibco Software.

## Related Research Documents

[The Future Of Mobile Application Development](#)

January 17, 2013

[Great Mobile Experiences Are Built On Systems Of Engagement](#)

November 16, 2012

[Cloud Mobile Development: Enabled By Back-End-As-A-Service, Mobile's New Middleware](#)

August 30, 2012

## MOBILE REQUIREMENTS LEAVE AGING WEB ARCHITECTURES OUT IN THE COLD

Great mobile experiences require a mobile app or optimized mobile website supported by what Forrester calls “systems of engagement” — robust back ends built on rich content, predictive analytics, cloud services, transaction systems, and personalization.<sup>1</sup> As an initial strategy to deal with the demands of mobile and connected products, companies have retrofitted their web servers or bolted content delivery network (CDN) shims onto a 15-year old web architecture designed for 17-inch monitors on a high-speed LAN connection. But it’s a recipe for disaster: New requirements are outpacing the kludgy workarounds of a web architecture. In the new mobile world, the velocity of releases, the performance unknowns of the last wireless mile, and a growing third-party collection of business and service partners will derail this old web-centric model, because it (see Figure 1):

- **Isn’t agile; the application layer remains hard-coded to the data and presentation tiers.** These old-style architectures exist as complete monolithic solutions, with a hard-coded dependency between the storage, delivery, and presentation of the data. As a result, updates require the drawn-out rebuilding of the entire application. Impatient business managers then have to wait for a slew of updates before IT can justify the cost of a new release. The hard-coded dependencies lead to 100-step product migrations that take many weeks to deploy — and each step carries a high risk of failure. In contrast, Amazon.com puts a code update into production every 11.6 seconds on average.
- **Can’t flex; they’re monolithic solutions, not a composition of best-of-breed offerings.** As a poor man’s workaround, companies bolt CDNs onto three-tier application server models. This doesn’t allow for fluid integration with third-party business partners or innovative external services like Google Maps. Modern apps and data are dynamic — cache solutions cannot be an afterthought. In parallel, content requirements continue to expand, exacerbating the problem: The size of the average web page has grown from 1.08 MB in 2012 to 1.50 MB in 2013!<sup>2</sup>
- **Doesn’t scale; services cannot rise and fall with changing demand.** Scaling today’s services involves deploying additional infrastructure such as network dispatchers or load balancers. This model rewards success with a larger maintenance bill. The definition of viral success — “Our online campaign was so popular that the servers crashed” — is also the definition of failed scalability. Giving salespeople access to back-end systems on mobile devices results in highly variable and elastic demand that also grows exponentially. Firms have plugged some of the holes in the three-tier dike by moving back-end systems such as customer relationship management (CRM) to the cloud, but this does not solve the scalability concerns of internally integrated systems such as business intelligence, role-based access, and content management.<sup>3</sup>

**Figure 1** The Current Web Implementation Is Broken

	Web architecture reality	Mobile engagement requirement
Agility	The application layer is hard-coded to the data and presentation layers.	Dynamically composed services that support contextual experiences
Flexibility	Monolithic solutions built from tightly coupled systems	Loosely coupled best-of-breed components to support mobile engagement
Scalability	Services cannot rise and fall with changing demand.	Distributed deployment for performance and rapid scaling

100161

Source: Forrester Research, Inc.

## Mobile Engagement Demands A New Set Of Capabilities

The new generation of mobile apps that emphasize serving people on any mobile device in their immediate context — what we call “mobile engagement” — place tremendous demand on your delivery architecture and design approaches.<sup>4</sup> The failures of the web-centric model will drive fundamental changes on four levels:

- 1. The challenges of wireless networks require atomized components.** The platform for mobile engagement has to be able to handle spotty networks, dropped connections, and long pauses between service requests.<sup>5</sup> To handle this demand, middleware must quickly assemble atomized and cached content, data, and app components on the device to keep things moving while the rest of the screen is loading. This allows a consumer on the subway to browse the initial content of your mCommerce site even if they lose their data connection. Trying to deliver the full content payload in one shot could give them no experience at all.
- 2. Elastic demand requires capacity that scales up and down at a moment’s notice.** Mobile app adoption and upgrade cycles are now measured in hours for millions of devices. Pioneers like USAA, HotSchedules, and Pandora have found that the number of activities (lookups, data downloads, and transactions) is five to 10 times higher on mobile than on the Web. These access patterns are becoming more volatile, which makes it too costly to handle the dynamic scale in your data center. To provide an elastic infrastructure, firms are experimenting with cloud-driven architectures that are more efficient and cost-effective.
- 3. The velocity of releases and feedback requires a modular design with continuous updates.** Adapting to user input drives success or failure in mobile apps. Reacting quickly to these criticisms requires software development agility to build apps on any mobile device and the ability to recompose the services that drive these experiences on the back end. In parallel, services like Facebook, Google Plus, Twitter, and Instagram have conditioned us to expect continuous application updates with no downtime or performance reduction during

deployment.<sup>6</sup> For the enterprise to follow suit, developers and operations staff must shorten the software development life cycle, which requires an architecture that allows developers to insert, update, or replace individual components without degrading the service at any point.<sup>7</sup>

4. **The aggregation of internal and external services demands open APIs.** Great mobile experiences integrate components from internal and external systems into simple, task-oriented apps. For example, Home Depot uses Twilio to send SMS messages to local contractors and Walgreens uses Urban Airship to send push notifications to its prescription customers. As third-party business partners or service providers expose their services via consumable application programming interfaces (APIs), your platform must be able to aggregate the growing pool of external resources — even though you only control your side of the API. When you're just the producer or consumer of information, you'll need to be much more flexible in how you partner with other digital providers.

## MOBILE NEEDS A FOUR-TIER ENGAGEMENT PLATFORM AND APPROACH

Throw out the old ideas of a three-tier architecture and embrace the new requirements for mobile engagement and connected products — and do this while preserving the investments you've made in services, service-oriented architecture, and web infrastructure. At Forrester, we believe that you can only deliver on the demands of mobile apps by deploying a modular four-tier architecture that we call an “engagement platform,” which we define as:

*An engagement platform supports a distributed, four-tier architecture natively engineered to deliver compelling experiences, excellent performance, and modular integration on any device over any network at Internet scale.*

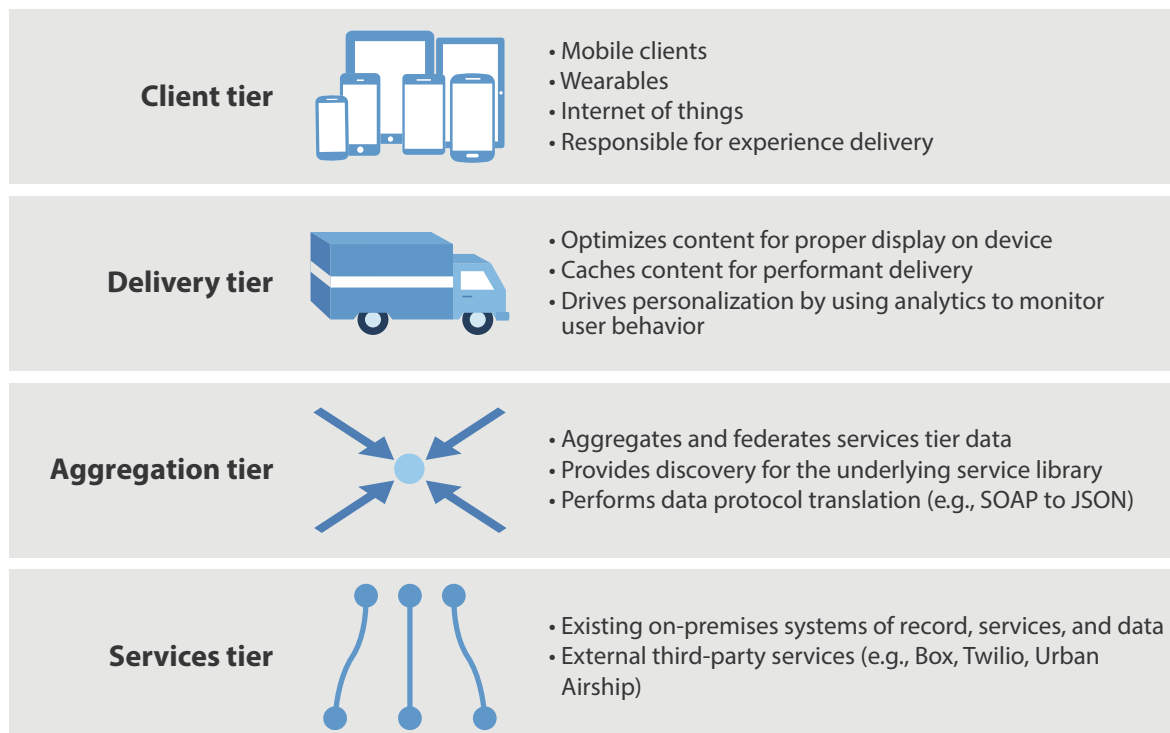
The core of this engagement platform is its fourth tier. By replacing a single application tier with two loosely coupled aggregation and delivery tiers, you will separate the hosting of data and content from its delivery. The result is a loosely coupled, four-tier engagement architecture (see Figure 2):

1. **A client layer accounts for the unique attributes of different devices.** This presentation layer insulates the unique capabilities of each app and device — desktop or mobile, browser or dedicated app — from the services that back-end applications deliver. This boundary allows developers to create back-end services like flight status and shipment notification independent of the mobile app that will consume them. Creating this clear boundary drives productivity for your developers without an onerous maintenance challenge; it will be critical for a fluid business partner network.
2. **A delivery segment handles special middle- and last-mile challenges.** This element uses intelligence from the client layer to determine the optimal way to deliver contextually appropriate content. The delivery tier accomplishes this by using over-the-wire content transformation — as

opposed to protocol-based conversions at the next aggregation layer — and leveraging edge-of-network cache functionality for increasingly dynamic data. CDNs such as those provided by Akamai, along with delivery optimization solutions like Instart Logic, application delivery controllers like Riverbed Stingray, and on-premises in-memory database caches, fulfill this today.

3. **An aggregation tier integrates internal and external services and transforms data.** This API layer has two brokerage roles, providing discoverability between app requests and services and bidirectional translation between client requests and back-end or third-party services. This makes composing the underlying data and services easier and enables relatively simple real-time translation to the appropriate data format. The service composition becomes more dynamic with the addition of business intelligence, analytics, and role-based access, which occurs in this tier.
4. **A services element spans internally and externally provisioned data and functionality.** This final architectural element dynamically composes data and business processes through a set of continuously deployable services. This tier provides data to the layers above without concern for how that data is consumed; the other layers can exist behind the corporate firewall or externally — or both! This allows for the ultimate flexibility in the consumption and dynamic composition of services, whether leveraged by apps or by the evolving partner ecosystem.

**Figure 2** The Four-Tier Engagement Platform Makes Delivery Its Own Tier



## Engagement Platforms Embrace New Design Principles

To get the most out of this new, loosely coupled four-tier architecture, developers and operations staff must embrace a shift in design thinking. Without a shift in approach, the old habits of tight coupling, service interdependencies, and rigid deployment models will put them right back where they started: inflexible, unscalable, and too slow to deploy. With a new design approach and a relentless commitment to logical separation, firms can design, build, and deploy components in isolation from the data and services those components consume and the experience they deliver. Application development and delivery pros must embrace and embody three new architectural principles:<sup>8</sup>

**1. Loosely coupled components that support wireless clients and aggregated services.**

Developers need to build pluggable, asynchronous architectures that will enable lightweight infrastructures that do not fail over transient networks. To do this well requires a shift in mindset to event-driven development. This means designing code that operates as disconnected, self-sustaining modules instead of the familiar monolithic chunks. The code and hardware independence that this doctrine creates support the elasticity required by spikes in demand. Adoption of this design philosophy is evidenced by the rise of nonblocking I/O services, such as nginx and Node.js, that handle each incoming request asynchronously.

**2. Distributed deployment for performance and rapid scaling.** Build only those components that differentiate your business instead of building everything in house. Developers must transition away from the “not invented here” mentality and toward one of “compose first, code last.” For instance, if your service needs to show a location on a map, don’t build it yourself; simply call the Google Maps API. This requirement to incorporate new capabilities will force you to adhere to good practices such as stable APIs to ensure separation of concerns, semantic translators to easily swap out a service, and continuous monitoring of external and internal service levels to look for bottlenecks or worrisome components.

**3. Dynamically composed services that deliver contextual experiences.** You need to atomize apps and services into bite-sized, sub-megabyte chunks to get the most out of middle- and last-mile caching solutions from Akamai, EdgeCast Networks, and others. Conduct code and system architecture reviews to emphasize your need for modular designs that create, reuse, and cache content. Granular content fragments allow mobile apps to dynamically compose this personalized content, which drives customer engagement.<sup>9</sup>

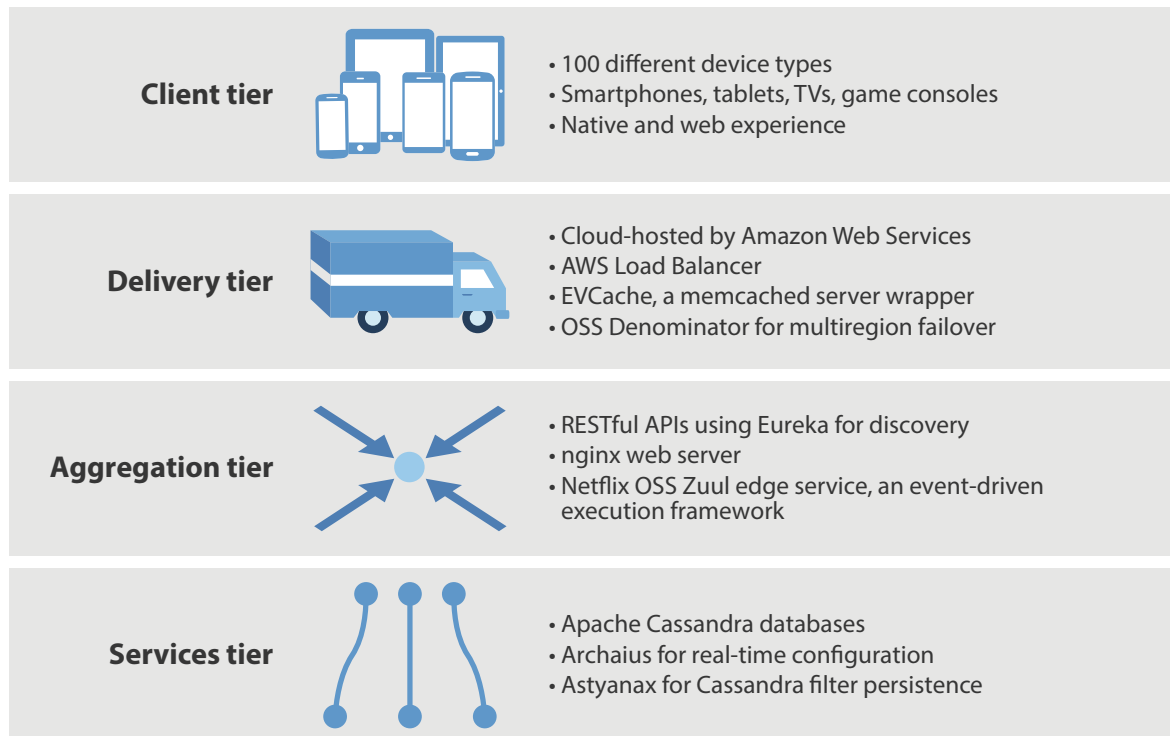
## EARLY EXAMPLES OF THE FOUR-TIER ENGAGEMENT PLATFORM

High-profile companies are experimenting with four-tier architectures and new design principles to deliver what customers need: great mobile experiences over any network. We found three examples that reinforce the architectural shift going on:

- **Netflix defines the need for performance and flexible deployment.** The Netflix API supports more than 1,000 different device types and handles more than 50,000 requests per second during peak hours. Netflix does this while actively deploying new features nearly every day. The firm focused on performance, scale, and continuous delivery when creating its current infrastructure, which it built largely on open source software (see Figure 3).
- **Kinvey provides an engagement platform architectural road map.** This back-end-as-a-service (BaaS) vendor provides a cloud solution for mobile enablement of corporate data.<sup>10</sup> It has published a reference architecture that acts as an early model of the stack, detailing a services tier, an aggregation tier, and a delivery tier that it delivers through its hosting solution.<sup>11</sup>
- **Salesforce.com's architecture hinges on a cloud aggregation tier.** This CRM groundbreaker drives innovation on the front end to maintain viability on the core back-end product. The aggregation tier is the central element of its migration to a four-tier architecture. To support multitenancy and front-end innovation, salesforce.com needed to abstract the underlying cloud and on-premises services while providing a consumable API layer and mobile libraries. Its enterprise mobile patterns contain a mobile platform pattern that details the recommended mobile architecture for front-end developers.<sup>12</sup> The platform brokers all communication between “native” client objects and back-end enterprise services. The enablement stacks are now flexible and modular; replacing or updating back-end services does not affect deployed clients, and the entire system scales elastically in the cloud.



**Figure 3** Netflix's Four-Tier Engagement Platform



100161

Source: Forrester Research, Inc.

## HOW THE ENGAGEMENT PLATFORM MARKET WILL UNFOLD

The transition to a four-tier engagement architecture is in its infancy. As vendors compete for platform supremacy, Forrester expects the market to develop in three stages through 2023:

- **Proprietary stacks (2009 to 2013).** So far, we've seen single-vendor solutions that attempt to operate according to the new architectural principles. BaaS offerings from AnyPresence, FeedHenry, Kinvey, and StackMob remain the initial leaders in an emerging market. API management offerings aggressively target the aggregation layer, positioning them as core to many successful four-tier solutions. Portal solutions, once in vogue for creating personalized, aggregated experiences, will pivot to focus solely on data integration.
- **Decoupled layers (2014 to 2017).** In the next phase, vendors will start to recognize their sweet spot and defensible market position in a specific tier. For example, IBM could reconceive its WebSphere product as an on-premises aggregation point that has direct and intimate, although loosely coupled, links with a next-generation CDN to handle content translation and caching.

Rather than owning the delivery and aggregation tiers, IBM may decide to focus on the aggregation and services tiers and let Akamai or Amazon.com own the delivery tier.

- **Aggregated platforms (2018 to 2023).** In this mature phase, major vendors will coalesce their platform services in a way that attracts an ecosystem of customers, developers, and supporting vendors. For example, Microsoft might preintegrate a CDN, ad server, location-based content translator, mapping service, analytics engine, merchandising service, and privacy assurance service into a “commerce service platform” hosted on Microsoft Azure. In this way, it can make it easier for retailers to build and operate a multitouchpoint experience. Microsoft won’t own all of the services, but it will offer them as a bundle.

### Three Vendor Camps Will Vie For The Aggregation Tier Sweet Spot

Part of the battle for supremacy will center on the critical aggregation tier, which defines the place where most of the code and content will be hosted and executed. Vendors from three very different categories will battle for ownership of the aggregation tier (see Figure 4):

- **Data integrators will use APIs to provide access to enterprise services.** Vendors like CA, Intel, and MuleSoft create a baseline of consumable content, exposing the underlying service layer through a set of discoverable, integrated, and federated APIs. Their advantage: They expedite development by providing a consistent interface to the many disparate back-end systems of record.
- **Middleware vendors will provide a full development environment for mobile.** Middleware vendors like IBM, Kony, and SAP currently provide an initial piecemeal implementation of the aggregation tier. They come at this opportunity with existing assets in application servers, application and device management, security, and directory integration, as well as a set of development tools. They have a large opportunity in that corporations already trust these vendors with their content. To capitalize on this, vendors will need to offer lightweight, flexible, cloud-driven solutions that do not lock customers into the complete software, hardware, and tooling stack.
- **Engagement platform pure-plays aim to provide best-of-breed aggregation.** These startups, including FeedHenry, KidoZen, and Kinvey along with SaaS pioneers like salesforce.com, are starting from scratch instead of refactoring an existing platform. They offer cloud-native solutions designed from the ground up for a four-tier model that supports continuous delivery, modular architectures, and elastic scale. This gives vendors the opportunity to fully enable an enterprise’s aggregation tier by providing consistent connectivity to systems of record while providing the flexibility to use best-of-breed tooling and third-party services.

**Figure 4** Vendor Firms Will Vie For Control Of The Aggregation Tier

	Data integrators	Middleware vendors	Early platform vendors
Description of aggregation tier focus	Provides API access to back-end data	End-to-end mobile integration and development solution built on existing platform	Encapsulate API access with mobile development libraries (both native and web)
Example vendors	<ul style="list-style-type: none"> <li>• Intel/Mashery</li> <li>• CA/Layer7</li> <li>• MuleSoft</li> <li>• Apigee</li> <li>• WSO2</li> <li>• IBM Cast Iron</li> </ul>	<ul style="list-style-type: none"> <li>• IBM MobileFirst</li> <li>• Kony</li> <li>• SAP</li> <li>• Oracle</li> </ul>	<ul style="list-style-type: none"> <li>• salesforce.com</li> <li>• Infor</li> <li>• AnyPresence</li> <li>• Kinvey</li> <li>• FeedHenry</li> <li>• KidoZen</li> <li>• Verivo</li> </ul>
Vendors' advantage	Expedites development by providing a consistent, consumable access layer	Trusted in the enterprise; solution includes development tooling	Built from the cloud down; allows best-of-breed tooling and service integration

100161

Source: Forrester Research, Inc.

## RECOMMENDATIONS

### TAKE FOUR STEPS TOWARD THE FOUR-TIER ENGAGEMENT ARCHITECTURE

Vendors are continuing down the path to the new four-tier model, but enough has already coalesced to make it possible for application development and delivery pros to start their own journey away from the three-tier Web and toward a four-layer engagement platform:

- **Use a BaaS provider for your high-traffic mobile apps.** When considering mobile app strategies, make use of an enterprise-class BaaS provider like AnyPresence, FeedHenry, Kinvey, or StackMob if an app will drive higher than usual traffic for your infrastructure and will require new services to support it. These vendors provide the aggregation and delivery layers of the architecture. Additionally, your mobile developers will see an increase in productivity; most vendors wrap access to back-end services in a developer-friendly native or JavaScript mobile library.
- **Integrate with external services.** Become craftsmen, not draftsmen: Transition from a development organization that excels at individual component creation to one that rewards the composition of existing services, building only when none are available. This requires an open, pluggable services tier as well as comfort with using open source software. Many of these services will exist in the cloud, so become adept at understanding cloud security models that work with federated identity. Aggregate these with context from mobile devices to provide an enhanced personalized experience while gathering valuable data on each user, helping you better meet their future needs. Do this initially by making use of analytics to personalize the service you offer to individual users.

- **Move to a lightweight, multifactor security model.** Yesterday's security models have a chatty design, session-based implementation, and require that each service that a user interacts with understands its own security context. Deploying this model on a four-tier architecture will bind the tiers together, defeating the goal of flexibility and modularization. Instead, make use of newer lightweight security mechanisms, such as SAML or OAuth, from vendors like CA Technologies, NetIQ, and Ping Identity.<sup>13</sup>
- **Architect solutions with an atomic services mindset.** You need to create stateless services that can exist on their own. Rely upon the aggregation tier to compose these services into an application response. Data will emerge as the new corporate currency; APIs are the brokers of this currency, so make sure that they are granular enough to encourage adoption by all types of clients. Deliver this data as fast as possible regardless of network conditions. This will require outside-in thinking to use the appropriate delivery tier technologies. Your implementation of the client tier will act as the road map and documentation that partners follow when consuming this data.

---

#### WHAT IT MEANS

### VENDORS WILL COALESCE AROUND THE TIER THEY CAN OWN

Over the next two years, platform vendors will begin to identify and self-organize around the layers that they can own. For example, IBM will extend WebSphere beyond Worklight to own the services and aggregation tier while partnering for the delivery tier. Akamai and next-generation CDNs will expand their base offerings to optimize delivery with regionally cached content while intelligently transforming that content for upcoming requests. We predict three major outcomes in this transition to the four-tier future of engagement:

- **Nginx and Node.js will become the dominant infrastructure for mobile and connected apps.** Servers that tie up critical computing power while processing the request/response HTTP cycle simply cannot scale efficiently enough for tomorrow's mobile demand. Event-driven, nonblocking infrastructure such as the nginx web server and the Node.js application platform operate on a single thread, brokering data between back-end servers and requesting clients. Nginx has already become the most popular web platform among the top 1,000 web servers in the world; together with Node.js, it will emerge as the de facto back-end implementation for engagement architectures.
- **Vendors like Amazon, Google, IBM, and Microsoft will gobble up mobile BaaS vendors.** Success patterns will appear as companies deploy back-end architectures for mobile that can perform at scale. The BaaS market has set the pace with this trend; vendors have optimized the aggregation and delivery layers and provided libraries to help create the

client tier. Initially, these suppliers will merge offerings with API management providers. Once this market has been established, global service vendors will acquire successful players to address these challenges for their clients.

- **Mobile engagement providers like Atos, SapienNitro, and Worldline will productize platforms.** Implementation vendors that are willing to stand up infrastructure to support mobile apps, such as SapienNitro's for European airlines or Atos' for McDonald's, have an opportunity to extend their offering in the form of reusable software platforms. Some of them may even move into the platform-as-a-service market to offer multitenant capabilities to compete with Amazon and Microsoft.

## SUPPLEMENTAL MATERIAL

### Companies Interviewed For This Report

Akamai Technologies	Kinvey
Antenna Software	Leapfactor
AnyPresence	Microsoft
Atos	New Relic
Cynergy Systems	salesforce.com
FeedHenry	SapienNitro
IBM	StackMob
Infor	Tibco Software
July Systems	

## ENDNOTES

- <sup>1</sup> Systems of engagement are different from the traditional systems of record that log transactions and keep the financial accounting in order: They focus on people, not processes. These new systems harness mobile, social, cloud, and big data innovation to deliver apps and smart products directly in the context of the daily lives and real-time workflows of customers, partners, and employees. The compelling notion of context — the sum total of what your customer has told you and is experiencing at the moment of engagement — is made possible with cloud delivery and predictive analytics applied to a blend of data from device sensors, social feeds, personal preferences, and systems of record. See the February 13, 2012, "[Mobile Is The New Face Of Engagement](#)" report.

- <sup>2</sup> Source: HTTP Archive (<http://httparchive.org/trends.php>).
- <sup>3</sup> Source: Scott Johnston, “Release Management Best Practices at Amazon,” Puppet Labs, July 5, 2013 (<http://puppetlabs.com/blog/release-management-best-practices-at-amazon>).
- <sup>4</sup> By 2014, more than 1 billion people will be using smartphones and tablets. To adapt to this new environment, your experiences must be part of systems of engagement. See the November 16, 2012, “[Great Mobile Experiences Are Built On Systems Of Engagement](#)” report.
- <sup>5</sup> Source: Dan Rowinski, “What is Slowing Down Your Mobile Apps?” ReadWrite, July 24, 2012 (<http://readwrite.com/2012/07/24/infographic-what-is-slowing-down-your-mobile-apps>).
- <sup>6</sup> Twitter went so far as to create a character, the “fail whale,” which it displays during service reductions or rare downtimes. This highlights the new model of service commitment: continuous delivery with zero consumer impact. Source: What is Fail Whale? (<http://www.whatisfailwhale.info>).
- <sup>7</sup> For additional discussion on the velocity of modern application releases, see the August 22, 2013, “[Development Landscape: 2013](#)” report.
- <sup>8</sup> For more detail on the shift to modern applications, see the January 17, 2013, “[The Future Of Mobile Application Development](#)” report.
- <sup>9</sup> For more detail on using context in mobile, see the July 11, 2011, “[The Future Of Mobile Is User Context](#)” report.
- <sup>10</sup> For more detail on the solutions offered by BaaS vendors, see the August 30, 2012, “[Cloud Mobile Development: Enabled By Back-End-As-A-Service, Mobile’s New Middleware](#)” report.
- <sup>11</sup> Kinvey has created an infographic for a mobile cloud reference architecture. Source: “Mobile Cloud Reference Architecture for Enterprise,” Kinvey (<http://www.kinvey.com/docs/kinvey-mobile-reference-architecture.pdf>).
- <sup>12</sup> Find details on the “Mobile Platform Pattern” as a part of the salesforce.com developer site. Source: developer.force.com ([http://wiki.developerforce.com/page/Enterprise\\_Mobile\\_Patterns](http://wiki.developerforce.com/page/Enterprise_Mobile_Patterns)).
- <sup>13</sup> For additional detail on lightweight identity and access management technologies, see the March 22, 2012, “[Navigate The Future Of Identity And Access Management](#)” report.

## About Forrester

A global research and advisory firm, Forrester inspires leaders, informs better decisions, and helps the world's top companies turn the complexity of change into business advantage. Our research-based insight and objective advice enable IT professionals to lead more successfully within IT and extend their impact beyond the traditional IT organization. Tailored to your individual role, our resources allow you to focus on important business issues — margin, speed, growth — first, technology second.

### FOR MORE INFORMATION

To find out how Forrester Research can help you be successful every day, please contact the office nearest you, or visit us at [www.forrester.com](http://www.forrester.com). For a complete list of worldwide locations, visit [www.forrester.com/about](http://www.forrester.com/about).

### CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866.367.7378, +1 617.613.5730, or [clientsupport@forrester.com](mailto:clientsupport@forrester.com). We offer quantity discounts and special pricing for academic and nonprofit institutions.



---

## Forrester Focuses On Application Development & Delivery Professionals

Responsible for leading the development and delivery of applications that support your company's business strategies, you also choose technology and architecture while managing people, skills, practices, and organization to maximize value. Forrester's subject-matter expertise and deep understanding of your role will help you create forward-thinking strategies; weigh opportunity against risk; justify decisions; and optimize your individual, team, and corporate performance.

« ANDREA DAVIES, client persona representing Application Development & Delivery Professionals