



Mobile Goes Back to the Future With SOA

As application development leans more and more toward developing mobile applications, architects can take comfort in still using SOA principles—just in a new way.

• EDITOR'S NOTE

• SOA PROVES
CRUCIAL TO MOBILE
DEVELOPMENT

• CACHE IN FOR
BETTER MOBILE
APP PERFORMANCE

• PUTTING MOBILE
IN CONTEXT

Take on Mobile Development With SOA

[HOME](#)[EDITOR'S NOTE](#)[SOA PROVES CRUCIAL
TO MOBILE DEVELOPMENT](#)[CACHE IN FOR BETTER
MOBILE APP
PERFORMANCE](#)[PUTTING MOBILE
IN CONTEXT](#)

LOOK TO THIS handbook for first-person accounts of mobile development projects that rely on SOA middleware stacks and principles. Enterprise architects featured in the following three stories stick to basic SOA principles, but the styles and tools they use are new twists on services like cloud, microservices and light-weight APIs.

Our lead story, by freelancer George Lawton, relates how a mobile advertising provider applies SOA principles to solve common [mobile device and application problems](#). Focused on online advertising performance, ClicksMob built a services framework for optimizing online traffic and managing socket interfaces. Lawton also details the SOA principles that can be used in mobile application projects.

The second story examines how DailyLook.com, an online shopping portal, improved mobile application performance by [designing a](#)

[SOA infrastructure](#) to use new caching services. Using a caching approach in a new middleware tier helped DailyLook's legacy Web and new mobile applications work well together and supported growing data volume.

In the final article, consultant Tom Nolle uncovers the new application design challenges inherent in meeting mobile application users' needs. His advice covers using SOA and REST principles to [build mobile contextual apps](#). He also discusses the effect this new style has on traditional application lifecycle management and performance.

While the SOA model is evolving, experts and users in these articles predict that legacy SOA principles will remain important in mobile computing and online business transaction environments. ■

JAN STAFFORD

Executive Editor, SearchSOA

SOA Proves Crucial to Mobile Development

ENTERPRISE ARCHITECTS NEED to consider a new style for applying traditional SOA principles to mobile application development. There is now a wide range of infrastructure and tools for designing mobile applications using the cloud, microservices and lightweight APIs. The most important uses for SOA in mobile development include architecting back-end API integration for accessing data and compute services, said Dave Linthicum, of consultancy Cloud Technology Partners.

“In essence, the mobile applications are made up of composite services, and thus the heavy lifting occurs on the back end,” Linthicum explained. “Architects also need to appreciate the role of good service-based design and not attempt to do too much within the application or partition the applications via the use of services. If services change a great deal over time, they should be put in their own domain.”

SOA has been moderately useful in getting

enterprise systems decoupled and interoperable. “The key value of SOA has been to enable the more agile construction of external facing systems of engagement with mobile and Web interfaces,” said Al Hilwa, an analyst at IDC.

MOBILE’S ARCHITECTURAL CHALLENGES

Guy Raz Nir, head of research and development at ClicksMob, developer of performance platforms for mobile advertising, said the company has spent considerable effort using SOA principles to address two emerging mobile development problems around the C10K problem and optimizing traffic for low-bandwidth, high-latency mobile devices.

The C10K problem refers to the management of socket interfaces for 10,000 simultaneous connections. With traditional Web applications, users connect, do their business and then move on. But with [mobile applications](#),

users often leave their applications connected to the back-end service. This means that enterprise architects need to optimize network sockets to handle a large number of clients at once.

Mobile clients are considered “more aggressive” because the scenarios they were designed for provide real-world, real-time responses, Nir said. In contrast, a desktop user accepts some faults, latency and inaccuracies that would discourage a mobile user.

ClicksMob has developed a Java- and Scala-based Spring Framework as the middleware tier to address these challenges. Some of their services are inherently Spring-oriented and rely heavily on Spring Web model-viewer-controller, Spring Data and others. Other facilities are simply based on pure Java, servlet containers and memcached-like systems to support zero-latency response.

EVOLUTION OF THE ESB

Traditional middleware grew complex and unwieldy because of too much dependence on enterprise services buses (ESBs), said Sachin

Agarwal, vice president of product marketing at Akana Inc., a SOA infrastructure provider. With the ESB, developers were encouraged to put more logic and intelligence into the central infrastructure and less into the services and the applications that consume them.

A lightweight middleware equivalent is still required, as it provides the connectivity layer between back-end applications and data sources. Agarwal believes there are four important mobile middleware uses for the new generation of lightweight mobile middleware:

- To connect mobile applications with back-end applications and data;
- To provide security mediation between mobile standards and back ends;
- To provide lightweight orchestration and mediation of protocols between mobile and back-end applications, as well as provide some business rule-based orchestration; and
- To provide auditing, logging, monitoring and analytics.

A good practice is to adopt the right combination of APIs, [mobile backend as a service](#) (MBaaS) and API management platforms to manage, publish, secure and orchestrate these APIs. This provides an optimal design pattern for connecting back-end data with mobile applications. Key components of this design pattern include a portal for developers; an API gateway to secure, orchestrate and optimize the mobile APIs; a back-end business application and data layer; and a monitoring and analytics

tier for these APIs.

Many classic ESBs came with these capabilities and encouraged developers to build more logic and intelligence into the middleware tier. The danger lies in creating a middleware tier that becomes more complex and bottlenecks agility.

Think about the principle of “less is more,” Nir said. “The less we incorporate technology and code, the better performance we provide.”

—George Lawton

Cache In for Better Mobile App Performance

[HOME](#)[EDITOR'S NOTE](#)[SOA PROVES CRUCIAL
TO MOBILE DEVELOPMENT](#)[CACHE IN FOR BETTER
MOBILE APP
PERFORMANCE](#)[PUTTING MOBILE
IN CONTEXT](#)

ENTERPRISE ARCHITECTS ARE finding they can improve mobile application performance through new approaches to crafting their SOA infrastructure and application development tooling. One approach is to build SOA infrastructure to use new caching services that support a larger percentage of Web application data. Another approach is to craft the infrastructure to support more efficient protocols for data synchronization.

“This was a good decision for us, and made it a lot easier to develop performant applications,” said Eric Marston, CTO of DailyLook.com, a shopping website that features daily fashion inspiration curated by professional stylists. It re-engineered its enterprise architecture and application development process with a comprehensive caching strategy to dramatically increase mobile app performance and reduce the load on its [back-end infrastructure](#).

The company previously invested a large

number of resources in a strong middleware tier so that Web and mobile applications could interface with the back end through common middleware components. This allowed for code reuse and improved application development, Marston said. But considerably more work was required to think about promoting better application cache ability.

THE MOBILE MISSION

Improving mobile application integration was critical for DailyLook. About 90% of the company's traffic comes from users of its native mobile app. Marston evaluated a number of [content distribution network](#) services but found that most of them were only optimized for static content, he said. Using Instart Logic's Software-Defined Application Delivery Service, DailyLook was also able to accelerate much of its JSON messaging between DailyLook's

back-end infrastructure and mobile clients.

The Web application existed before the mobile app, so DailyLook refactored the Web application to make better use of this new middleware tier. This has made it easier to work with the Instart platform. Marston recommends that organizations consider organizing their middleware tier to cache JSON data objects. This makes it possible to stage data at the edge along with everything else.

CACHE SUCCESS

“This has reduced our database loads, Web server loads, and the application responds much quicker,” Marston said. Getting there can be a challenge for many organizations, but it is essential for enterprises in the retail space. The process involved figuring out how to segment and pull the data out so that anything personalized or specific to individuals was separated from the more static product data. In some

cases, this meant two smaller calls were made to the DailyLook infrastructure, but it also meant that more data could be staged closer to the users.

Now users are able to receive updates from the edge servers in about 30 milliseconds rather than the previous 500 milliseconds. Overall, this strategy has allowed DailyLook to offload about 70% of its total data requests to the edge servers.

Another good practice is developing efficient, business-specific caching rules, Marston said. Data relating to popular new style requests can be cached for several hours or even days, but other data is far more transitory.

Marston recommends that organizations encourage developers to consider better caching strategies during code reviews. This helps ensure that new applications are optimal.

Developers should also consider caching strategies as part of the middleware architecture. DailyLook was about halfway through

Organizations should encourage developers to consider better caching strategies during code reviews, helping ensure applications are optimal.

deploying its middleware tier before realizing it could be more efficient with a good caching strategy. Marston also recommends using a technology like JSON that is easy to cache. JSON is not as easy to use as HTML caching, but it can help reduce the load on database servers.

CONSIDER NEW PROTOCOLS

Other mobile experts are encouraging developers to think about data synchronization as a more efficient alternative to caching. “The real value is when the data synchronization system is lined up with all the other parts of the application stack,” said Matt DeBergalis, co-founder of Meteor Development Group, which makes the Meteor open source JavaScript development framework.

For example, the Meteor development platform supports direct integration with the Distributed Data Protocol that uses WebSockets as a more efficient protocol than REST or JSON in many cases. In Meteor, when the server updates information, the screen on the mobile device automatically updates. The developer

does not have to write special code to take advantage of automated synchronization. The end result is that data is cached on the mobile device rather than in the middle.

The HTTP protocol, which is traditionally used for [JSON and REST](#), is oriented toward retrieving documents from the Web, but live-updating apps need a way to send updates without the client asking. More high-performing apps can use emerging new protocols, like WebSockets, that are capable of automatically sending data to the client without a specific request. But the basic WebSockets protocol is low-level, which makes it difficult to use efficiently in applications, DeBergalis said.

DESIGN FOR SYNCHRONIZATION

The biggest challenge with building these kinds of apps is in the data layer on the back end. All databases to date have been designed to respond to queries, not to push data out to the back end when something changes, said Slava Akhmechet, founder of ReThinkDB, an open source database synchronization tool provider. Back-end engineers are forced to build complex

infrastructures and sophisticated custom code to work around these limitations in the data layer. Much of this work requires custom code on the front end, which further complicates the development and deployment process.

Sophisticated teams with big budgets and lots of time can implement this functionality on top of existing databases using a combination of traditional features such as views, triggers and third-party queuing systems like RabbitMQ. That can work quite well, but it is challenging and expensive to build because this approach is error-prone and takes a lot of development resources, Akhmechet noted.

“In the last year or two, there’s been tremendous innovation on the front end to make building real-time apps like these possible,” Akhmechet said. Frameworks like Meteor,

Angular and React treat real-time uses as first-class citizens, and all major Web browsers support technologies like WebSockets and

“There’s been tremendous innovation on the front end to make building real-time apps like these possible.”

—SLAVA AKHMECHET, founder,
ReThinkDB

long-lived HTTP connections to make pushing data to the browser in real time fast and easy. These tools promise a viable alternative to traditional caching networks when creating the next generation of mobile apps, Akhmechet said. —George Lawton

Putting Mobile in Context

EXPLOITING MOBILE DEVICES to empower workers is the next frontier in productivity enhancement for most companies. These new apps often demand different design practices, including interfaces and component models. In many cases, they involve Web-based front-end elements based on RESTful principles.

At the same time, these productivity-enhancing apps may involve existing components that use SOA principles. To make this mixture work, architects should use tiered app principles to present information, examine context and state control carefully and be prepared to rethink some interface and application lifecycle management (ALM) decisions.

In [mobile contextual apps](#), workers should expect to use a mobile device on scene to get information on or assistance with a specific task. In nearly all cases, information will be available through traditional apps, but the circumstances of the mobile worker make

accessing it in the normal way difficult or inefficient. Comparatively, mobile contextual applications are based on information services that can structure delivery according to the limits of the devices, limit the need for navigating menus and bridge traditional application lines.

A CAREER IN MODELING

The logical model for these applications is an agile front end that is device- and context-aware, drawing information from existing app components but not replacing the apps the components were created for. Thus, mobile contextual services are cross-grain to current application processes, including ALM.

The point of using contextual applications is to increase flexibility and agility without breaking current models. This can be done using Web technology to create a GUI and app

[HOME](#)[EDITOR'S NOTE](#)[SOA PROVES CRUCIAL
TO MOBILE DEVELOPMENT](#)[CACHE IN FOR BETTER
MOBILE APP
PERFORMANCE](#)[PUTTING MOBILE
IN CONTEXT](#)

portal that will insulate current apps and components from changes.

Tiered apps are common, but mobility and contextual services introduce two issues: worker context and performance at scale. Context often comes in part from the mobile device (location-based services) but will also require some worker input. Performance at scale means defining clear boundaries for delay and availability and ensuring these are met even when large numbers of workers are using an application. The latter is tricky because it's difficult to predict the precise number of simultaneous points of activity.

Virtually all mobile contextual services will define a specific sequence of expected tasks and have to be followed with minimal worker interaction. The expanded version can be maintained in the device or the application. Either mechanism is suitable, but where state information is voluminous, device-kept state can increase mobile network traffic and delay.

MANAGING PERFORMANCE

State control in the app has disadvantages, too. Software components that have to maintain state for users are more difficult to replicate. Take care in designing the scale-in and scale-out of Web-based front-end tasks to ensure state information is not compromised. This often means state or context control at the app level should be applied to the gateway component between the Web-based front-end processes of the app and the legacy SOA-based application flows.

It may be best to visualize two levels of load balancing and scaling when using back-end state control. The first, involving front-end Web-based components, would be stateless or rely on state or context from the mobile device. The second, the on-ramp to the actual application components, would employ back-end state control. To avoid having to distribute state databases across a wide geography, front-end components could be assigned a specific on-ramp.

The point of using contextual applications is to increase flexibility and agility without interfering with pre-existing applications.

The combined set of front-end and gateway components become virtual users of existing app components, likely linked with one another other via [SOA versus REST](#). The mobile contextual services are often different from normal

Mobile and contextual services are almost event-driven in their flows, which may cause traditional ALM to lose relevance.

workflows and may not use existing service bus logic for steering and sequencing. If they do, they may have to be adapted in the business process execution language (BPEL) to accommodate ad hoc mobile or contextual inputs and normal workflows through the same logic.

Dualism may break current interface models, either suggesting current componentization plans be changed to accommodate mobile and contextual use, or that new information elements be added to the data package passed to

a component. If component data models are changed, modifications in the BPEL will need to be made for legacy workflows to limit the impact.

The other area of impact is ALM. Because mobile and contextual services are almost event-driven in their flows, traditional ALM may lose relevance. This will likely place greater emphasis on unit-component behaviors for each app element and separate testing or certification of workflows and worker contexts. This is possible if you can think of a worker's activity as personal workflow, but it will likely be necessary to coordinate changes in operational practices with ALM to ensure all scenarios are covered.

If mobility becomes the rule in [worker empowerment](#), current efforts to accommodate mobile contextual services may be the entry point for driving major changes in application design. Keep this in mind when adding mobile worker support and preparing for a mobile-driven future. —Tom Nolle

GEORGE LAWTON is a journalist based near San Francisco. Over the last 15 years, he has written over 2,000 stories and steadily contributed to TechTarget's application development sites. Email him at glawton@gmail.com or check out his [website](#).

TOM NOLLE is the president of CIMI Corp., a consultancy specializing in telecommunications and data communications since 1982. He writes for many TechTarget websites. Read his [blog](#) or email him at tnolle@cimicorp.com.

STAY CONNECTED!



Follow [@SearchSOA](#) today.



Mobile Goes Back to the Future With SOA
is a [SearchSOA.com](#) e-publication.

Jason Sparapani | Managing Editor

Moriah Sargent | Associate Managing Editor

Jan Stafford | Executive Editor

Brein Matturro | Site Managing Editor

Linda Koury | Director of Online Design

Neva Maniscalco | Graphic Designer

Doug Olender | Publisher | dolender@techtarget.com

Annie Matthews | Director of Sales
amatthews@techtarget.com

TechTarget
275 Grove Street, Newton, MA 02466
www.techtarget.com

© 2015 TechTarget Inc. No part of this publication may be transmitted or reproduced in any form or by any means without written permission from the publisher. TechTarget reprints are available through [The YGS Group](#).

About TechTarget: TechTarget publishes media for information technology professionals. More than 100 focused websites enable quick access to a deep store of news, advice and analysis about the technologies, products and processes crucial to your job. Our live and virtual events give you direct access to independent expert commentary and advice. At IT Knowledge Exchange, our social community, you can get advice and share solutions with peers and experts.

COVER ART: FOTOLIA