Disposable Infrastructure for Disposable Apps

Lori MacVittie, 2015-25-05

Conferences agendas. Event navigation. Specific tasks, like buying a house or getting a car loan.

If you've installed an app for any of these things you've installed what's known as a "disposable mobile app" or DMA. Apps designed for a single use-case and with the expectation they'll be "thrown away" like brochures. Deleted until needed again. These apps are necessarily small, agile and highly volatile. Sometimes existing only for a short time - say to support an event like an election, the World Cup or a music festival - or existing for a long time on the "server" side, but not the client, like navigating a home mortgage process.

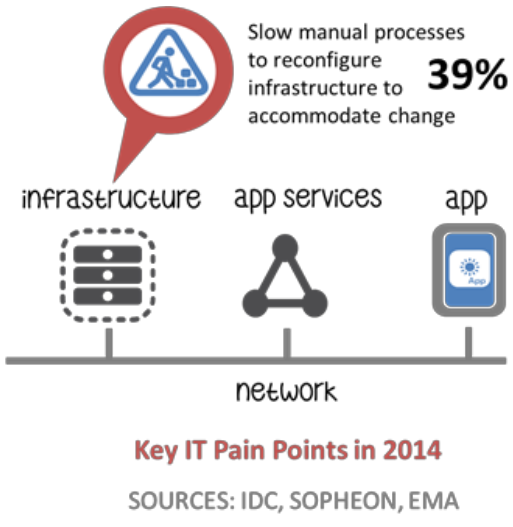
These apps are increasingly popular and are considered a kind of "micro app" (which is very similar to microservices but designed for third-party, not internal, development).

The reason it's important to note the existence of these kinds of apps is that they may have very different lifespans. From cradle to grave they may only exist for days, weeks or months. They flare into existence and then just as quickly they fade away. They are disposable, for the most part, and thus the infrastructure supporting them is also likely disposable.

In an age of virtualized data centers, with software eating IT, this notion is not as crazy as it once may have sounded. It's not like we're tossing out multi-thousand dollar hardware, after all. It's just some bits that can be easily put up and torn down with the click of an enter key.

That might be good if your only infrastructure concern is a web/app server. But as it happens these things need scalability and help with performance (they're mobile apps, after all, with most of the processing done on the server - cloud - side) and that implies several pieces of what's generally considered network infrastructure. Load balancing and caching and optimizing services. That means they, too, need to be "disposable". They must be software (or virtual) deployable and come with a robust set of APIs and templates through which the things can be quickly provisioned, configured and later torn down. It also means they must be cloud-ready or cloud-enabled or cloudified (whichever marketing term you prefer to be applied) so that the infrastructure and services supporting these disposable apps are as flexible and disposable as the applications they're delivering.

It also means a DevOps approach is increasingly important to managing these very volatile environments in which many more apps are delivered and disposed of in shorter cycles. A single, monolithic "one app fits all our offerings" approach is not necessarily the way disposable mobile apps are conceived of and delivered. They are focused and purposeful, meaning they are focused on providing a specific set of functionality that will never be extended. Other functions and purposes then are delivered via other apps, which increases overall the number of applications necessary and puts additional pressure on operations to deploy and deliver those apps. Each of those apps has specific services, configurations and monitoring requirements that must be tailored to the app. One size does not fit all in the world of applications. That's a marked difference from a world in which infrastructure configuration may remain largely unchanged for the lifetime of the app excepting performance or security tweaks. That means more work, more complexity, more often for operations who must manage the infrastructure.



That's why it's increasingly important for infrastructure to not only be software or virtualized, to not only present a robust provisioning and configuration API, but to also focus on participation in the growing automation ecosystems of popular frameworks and toolsets like Puppet and Chef, VMware and Cisco, OpenStack and Salt Stack. These are the frameworks enabling continuous delivery to leak out of dev and into operations and provide the means by which infrastructure is as easily put into production as it is disposed of.

Software is eating IT, but that should be taken as a *good* thing. DevOps as an approach to lifecycle management across the application and infrastructure spectrum is necessary to manage the growth being driven by the software eating the world. CA and Vanson Bourne found quantifiable benefits of adopting a DevOps approach in a global survey, with 21% of respondents reporting more new software and services were possible and 18% saw a faster time to market.

That kind of agility and speed is a requirement if you're going to be supporting disposable apps and the disposable infrastructure needed to deliver them. A key IT pain point is the reality that the network is still in the way. According to EMA research, "slow manual processes to reconfigure infrastructure to accommodate change" was cited by 39% of organizations as a significant pain point in 2014.

Applying DevOps to "the network" will aid in eliminating this significant point of impedance on the path to production. Part of that effort includes disposing of our preconceptions about the nature of the network (it's hardware! It's untouchable! It's not my domain!) and start considering which pieces of the network are ripe for being treated as disposable as the apps it delivers.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com