10 Predictions for Developers in 2019

By Siddhartha Agarwal

1. Eight out of 10 custom applications will include embedded intelligence as a first-class capability by 2020.

Developers need to understand how data scientists---who write code, but think in math and models---and engineers will understand elements such as how to employ build and deploy their cloud native applications. Platform engineers will marshal all the resources needed to have a high performance cloud infrastructure on which developers can develop and deploy their cloud native applications. Platform engineers will understand elements such as how to employ Kubernetes and Envoy proxies to run a service mesh, which is increasingly important for the complex service-to-service communication of microservice applications.

They'll also have an emerging necessity of containers, and conversation between services, since both security and availability are essential to using containers at scale within the enterprise. Developers will get the benefits of the cloud---they can develop and deploy on containers, and the cloud will scale resources up and down to serve first-class apps based on demand. But if a platform engineer who will pull all that infrastructure together and ramp it up for developers to use.

2. Developers will need to partner with a platform engineer, which will emerge as a key new role for cloud native development.

Think of this new platform engineer role as something like the IT admin for the cloud. The platform engineer will manifest all the resources needed to have a high performing cloud infrastructure on which developers can build and deploy their cloud native applications. Platform engineers will understand elements such as how to employ Kubernetes and Envoy proxies to run a service mesh, which is increasingly important for the complex service-to-service communication of microservice applications.

3. To balance security and performance needs, a hybrid model that falls between virtual machines and containers will rise in popularity for deploying applications.

The past year brought some early security threats that would span several developers that were sitting on platforms that they got security wrong. In response, look for an application deployment approach that applies the right cloud and service mesh for the performance needs of the performance and efficiency of the performance and efficiency of the container.

4. The economics of serverless become too compelling to ignore, driving serverless innovation on multiple fronts.

For starters, an open standard for serverless will take clear shape, allowing developers to embrace serverless like they have kubernetes, without the lock-in risk many of today’s serverless options have. But non-prod will probably exactly how that will take shape — one leading platform, or several platforms with common, standards-based building blocks? But it won’t be major projects to this time next year. Without serverless, solutions will quickly extend from today’s function as a service providers will drive this innovation, but developers will need to understand these options and what performance tradeoffs and security gains they can bring.

5. As bots proliferate, developers turn to ‘assistants’ to find the right bot for the job.

Developers are going to write lots of bots, with each one representing some kind of task—talking in order, looking up inventory, or scheduling a delivery. But those can’t be expected to run in the right tool for every job. Instead, users need an intelligent interface—a digital assistant—that can look at each possible task, and recruit it as needed to get a task done. So if you go into your HR app, you can

---

**Design: Pages 1-3**
6 Developers choose their clouds based on openness, and reject cloud lock in:

Developers move to cloud infrastructure so they can focus on creating, rather than on IT operations. So when they pick a cloud to build on, they need a choice of languages, databases, and compute shapes. And they want to be able to move workloads to any cloud if they're unhappy with their current provider.

A new fog of forked opensource that limits the ability to move among clouds, or the ability to adopt new innovations is an unhelpful barrier. That's why open source and open standards will be a core role as developers choose clouds that avoid cloud lock in.

7 Developers decide one cloud isn't enough.

While the first wave of cloud often leaned on a single provider, the current migration will be a multi cloud strategy, especially with the plethora of SaaS applications today. With a current expansion will be a multi cloud approach, where developers can pick a cloud to build on, they want a choice of languages, databases, and compute shapes. And they want to be able to move workloads to any cloud. They're increasingly wary of forked open source that limits the ability to move among clouds, or the ability to adopt new innovations. A new fog of forked opensource that limits the ability to move among clouds, or the ability to adopt new innovations is an unhelpful barrier. That's why open source and open standards will be a core role as developers choose clouds that avoid cloud lock in.

8 Autonomous Databases will let developers speed their applications to market scale:

Developers want more than rapid prototyping. They also want, when that prototype hits the mark, to get as many people using their work, as fast as they can. That's what solidifies success — and lets developers get feedback to make an application better. A decoupled, autonomous database makes both speed and scale possible. At a autonomous database, a developer doesn't have to think about scaling, patching, provisioning, or tuning a database. Instead, they can just focus on building their app, knowing they have a repository for their data that will automatically scale to meet their needs. Developers gain productivity since they can launch a high-powered database in minutes without help from a DBA. Scale comes from within the database, so developers can focus on creating, rather than on IT operations. And they can focus on creating, rather than on IT operations. So when they pick a cloud to build on, they need a choice of languages, databases, and compute shapes. And they want to be able to move workloads to any cloud if they're unhappy with their current provider.

9 Legacy, enterprise applications jump to cloud-native development approaches:

Developers working on applications that draw data from a blockchain ledger can deliver greater trust and transparency across cloud and on-premises deployments. An API that delivers blockchain transactions can now be the building blocks of applications that are used in regulated industries such as food and agriculture, pharmaceuticals, or finance. Developers of cloud applications can now build a customer-facing app using blockchain technology, such as assuring the performance and scalability. Why blockchain, and not a database? Blockchain can deliver greater trust and transparency across cloud and on-premises environments. And blockchain can let developers build on the skills they have and extend them into cloud-native environments.

10 If you're a developer in a regulated industry, blockchain is coming your way.

Developers working on applications that draw data from a blockchain ledger can deliver greater trust and transparency across cloud and on-premises deployments. An API that delivers blockchain transactions can now be the building blocks of applications that are used in regulated industries such as food and agriculture, pharmaceuticals, or finance. Developers of cloud applications can now build a customer-facing app using blockchain technology, such as assuring the performance and scalability. Why blockchain, and not a database? Blockchain can deliver greater trust and transparency across cloud and on-premises environments. And blockchain can let developers build on the skills they have and extend them into cloud-native environments.

Oracle | 10 Predictions for Developers in 2019

Try Oracle Cloud for Free

Get Oracle Cloud now > Visit developer.oracle.com >

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. Oracle, Oracle Cloud, and other Oracle and/or method names are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Any use of Oracle or Oracle Cloud identifiers is subject to the Oracle Cloud terms of service. Oracle Cloud is not available in all countries/regions. Visit developer.oracle.com for more information. Oracle and/or its affiliates. All rights reserved.
Thank you!