



DEVELOPERS LIVE

AEM as a Cloud Service Local SDK Build Analysers

Brian Chaikelson | Principal Product Manager

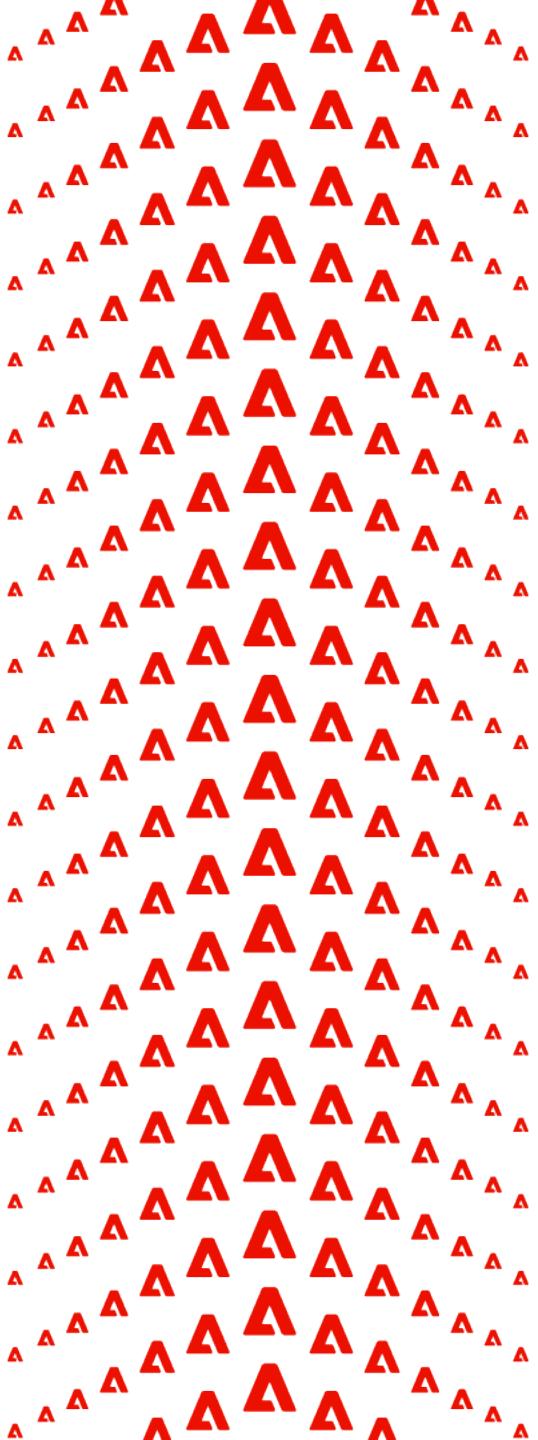
David Bosschaert | Senior Software Engineer

Karl Pauls | Computer Scientist

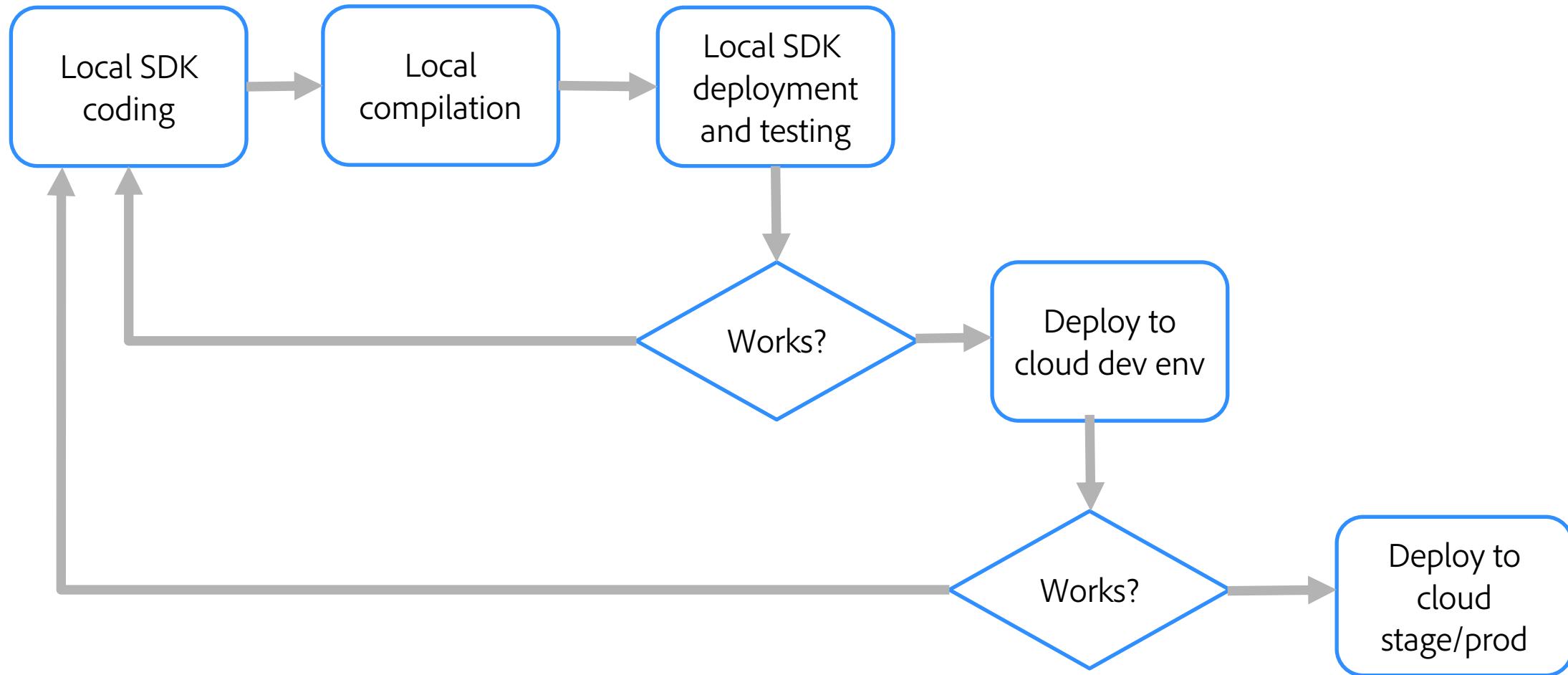


Agenda

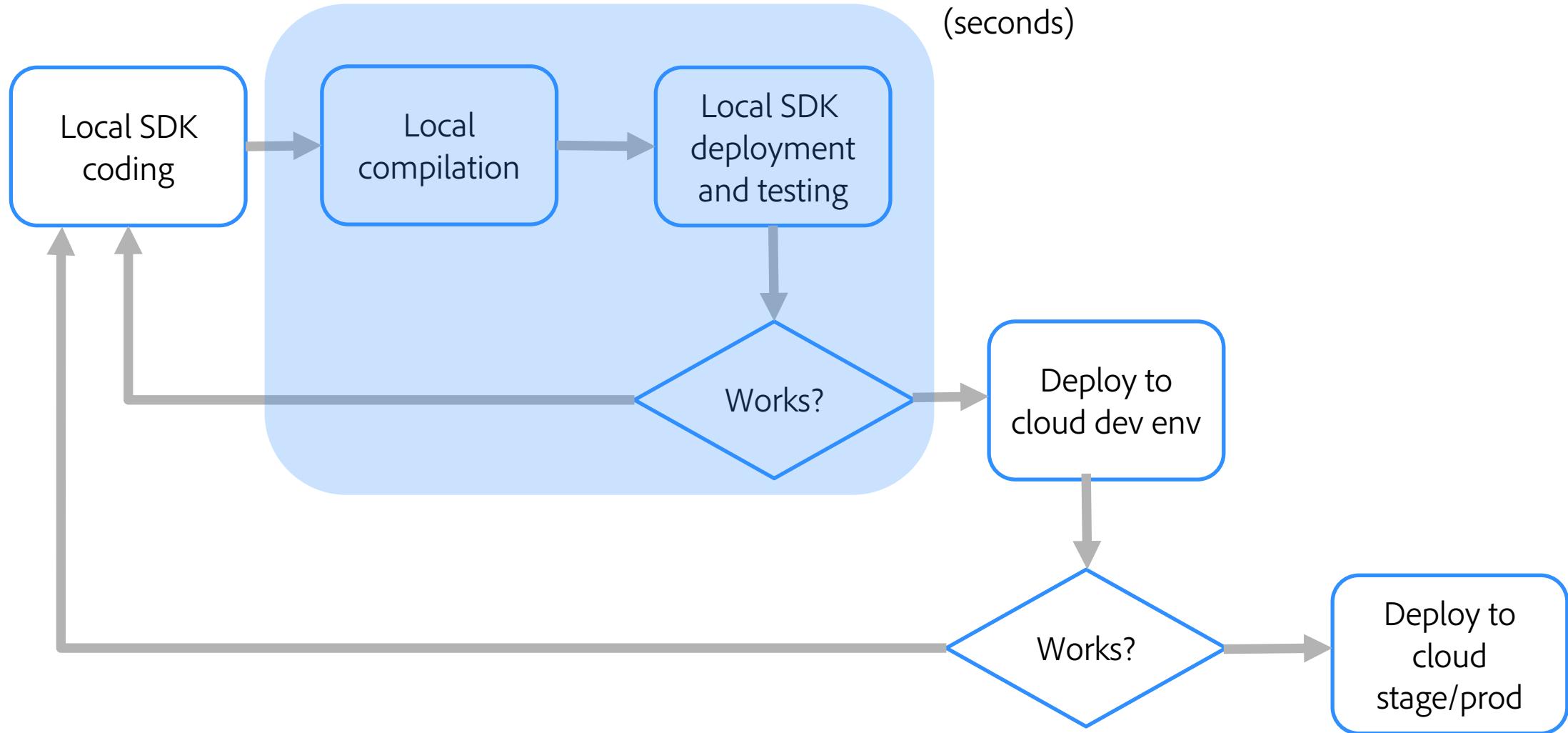
Brian Chaikelson	Background and motivation
David Bosschaert	Analyser details
Karl Pauls	Demo
Q&A	



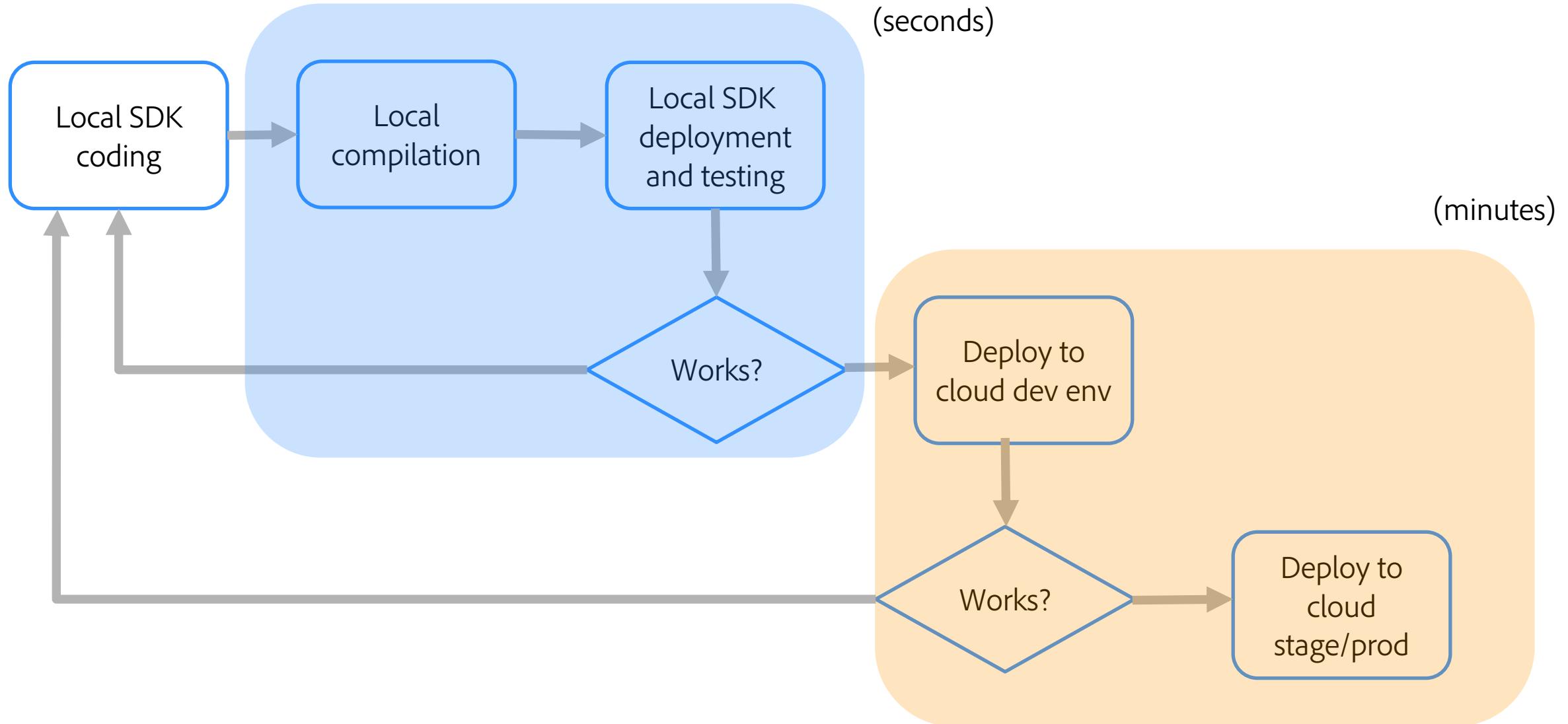
AEM as a Cloud Service Developer Flow



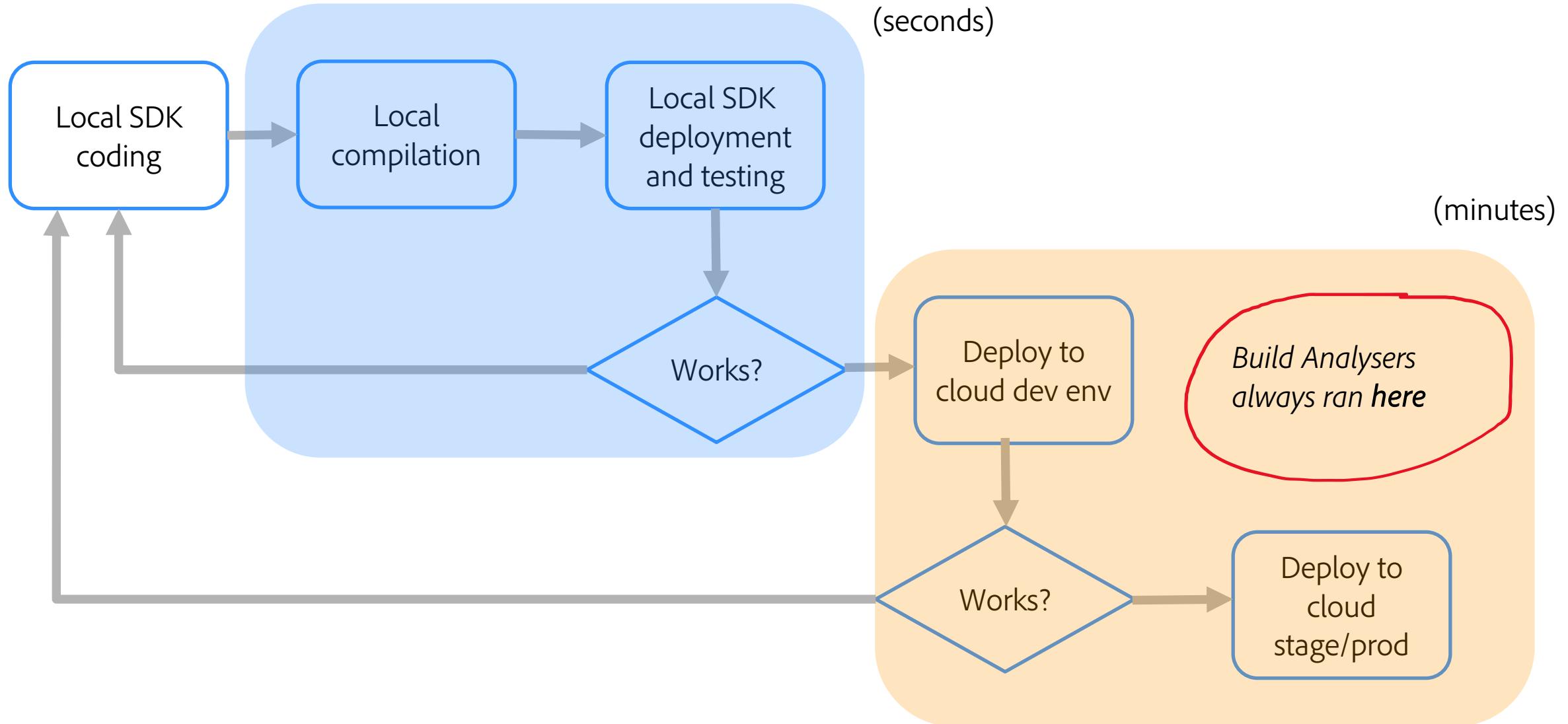
AEM as a Cloud Service Developer Flow



AEM as a Cloud Service Developer Flow



AEM as a Cloud Service Developer Flow



AEM as a Cloud Service Developer Flow

CI/CD orchestrated via Cloud Manager,
with steps and quality gates:

1. Validation
2. Build & Unit Testing
3. Code Scanning
4. Build Images
5. Deploy

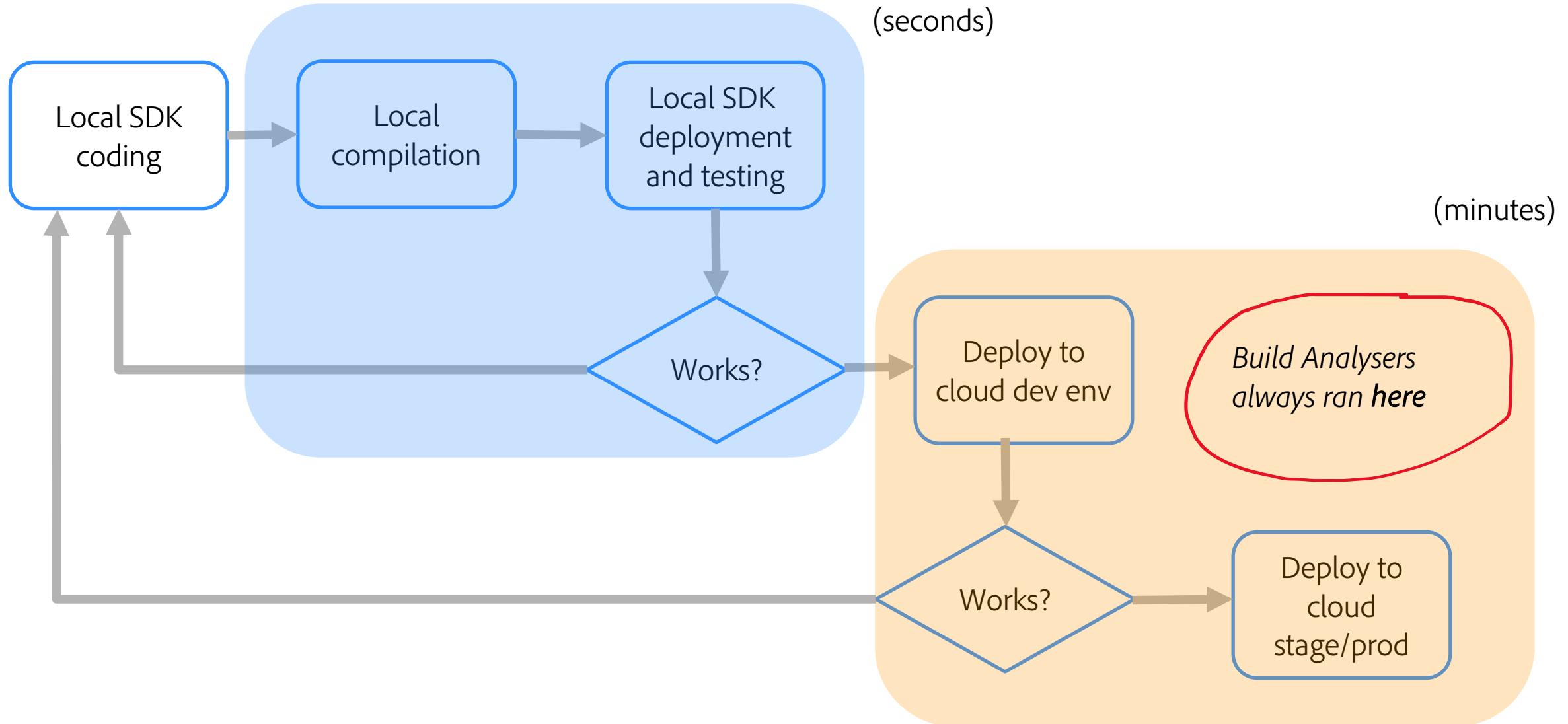
Analyser looks for issues in the maven project.

The screenshot shows the 'Program Overview' section of the Cloud Manager interface. At the top, it displays 'Brian's Standard' under 'Program Overview' and 'Non-Production Pipeline' with a 'Finished' status. The pipeline consists of several steps:

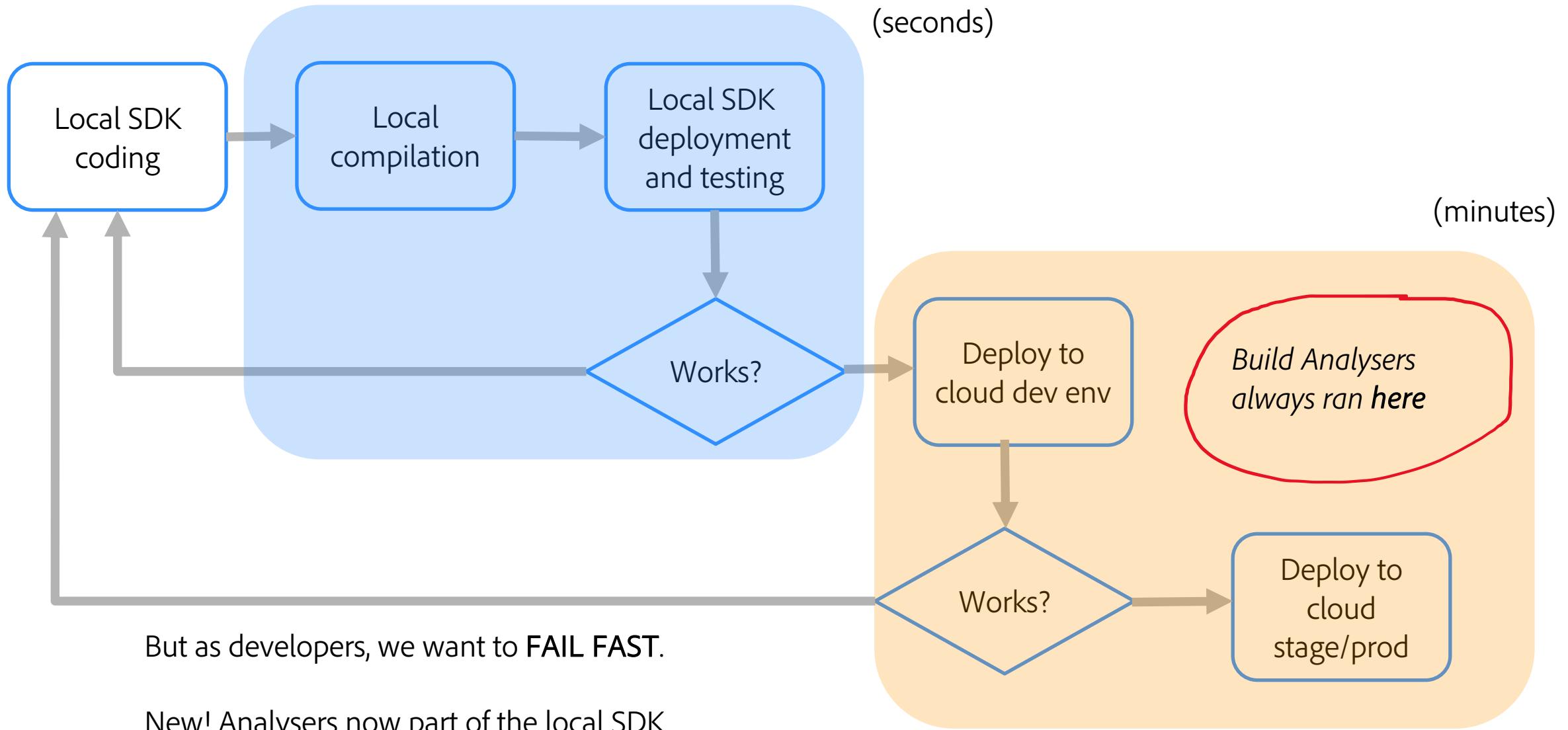
- Build and Code Scanning**: Includes 'Non-Prod Deployment'.
- Validation**: Status: Passed (Finished: January 15, 2021 3:50:56 PM PST).
- Build & Unit Testing**: Status: Succeeded (Finished: January 15, 2021 4:01:15 PM PST). Actions: Download Log.
- Code Scanning**: Status: Passed (Finished: January 15, 2021 4:01:22 PM PST). Actions: Review Summary, Download Details, Download Log.
- Build Images**: Status: Completed (Finished: January 15, 2021 4:20:22 PM PST). Actions: Download Log.
- NON-PROD DEPLOYMENT**: Includes 'Deploy to Dev'.

Details for the 'Build & Unit Testing' step mention a branch named 'wknd'. The 'Code Scanning' step notes an AEM RELEASE: 2020.12.4687.20201218T031154Z.

AEM as a Cloud Service Developer Flow



AEM as a Cloud Service Developer Flow



AEM Analyser Maven Plugin

Runs the AEMaaCS analysers in your local builds

Analyser	Function
api-regions-exportsimports	Checks if all OSGi bundles have their Import-Package declarations satisfied by the Export-Package declaration of bundles
bundle-nativecode	Ensures that bundles don't contain any native code
api-regions-crossfeature-dups	Checks that customer OSGi bundles don't have Export-Package declarations that override AEM as a Cloud Service's public API
repoinit	Checks the syntax of all repoinit sections
... other ...	Additional analysers perform internal validations

Analyser Framework

- Local Analysers
 - executed via the aemanalyser-maven-plugin [0]
 - Simple: create a new submodule to execute the analysers
- Analyser framework is open source [1]
- You create your own analysers too and add them ☺

[0] <https://github.com/adobe/aemanalyser-maven-plugin>

[1] <https://github.com/apache/sling-org-apache-sling-feature-analyser>



Image: Adobe Stock

Enabling the local analysers

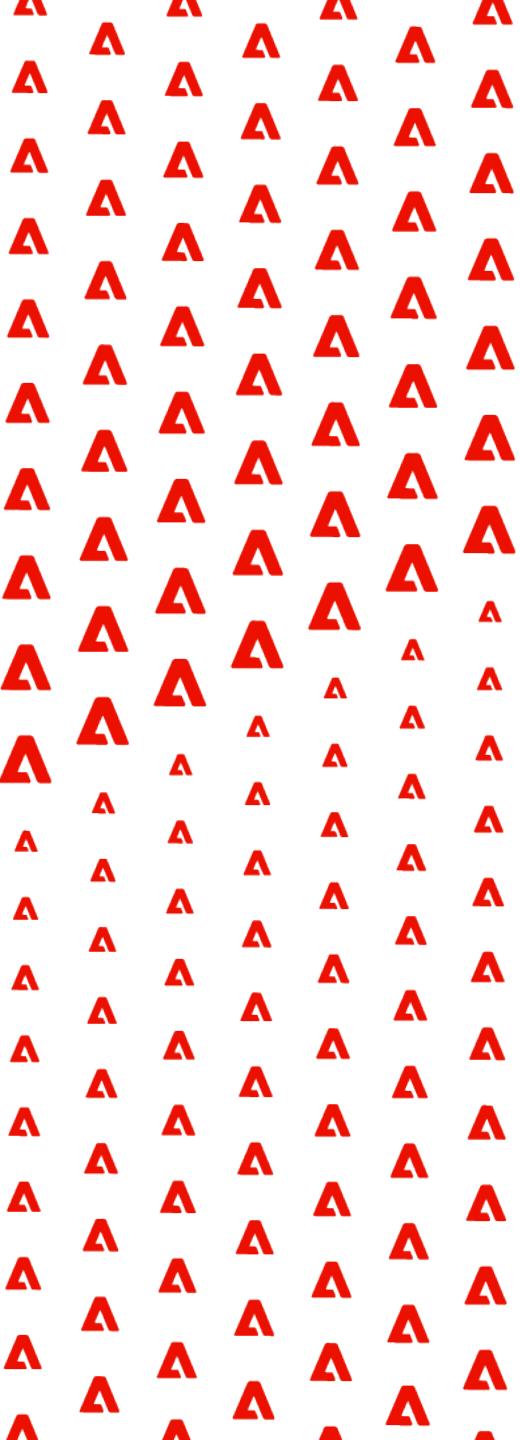
```
<project>
  ...
  <packaging>aem-analyse</packaging>

  <build>
    <plugins>
      <plugin>
        <groupId>com.adobe.aem</groupId>
        <artifactId>aemanalyser-maven-plugin</artifactId>
        <extensions>true</extensions>
      </plugin>
    </plugins>
  </build>

  <dependencies>
    <dependency>
      <groupId>com.mysite</groupId>
      <artifactId>myproject.all</artifactId>
      <type>zip</type>
    </dependency>
  </dependencies>
</project>
```

Make sure com.adobe.aem:aem-sdk-api is available in dependency management - AEM Archetype does this also for you.

Demo



Takeaways and Q&A

Use local SDK build analysers to detect problems in your maven project locally to FAIL FAST in seconds, rather than deploying to Cloud, which takes minutes.

Include them in your pom.xml today!

Public documentation: <http://adobe.ly/39RpVFs>

