



Large Scale Continuous Localisation Design

Considerations and mechanics for enterprise organisations

Gary Lefman

Chartered Engineer

2020-08-22



@CiscoL10N



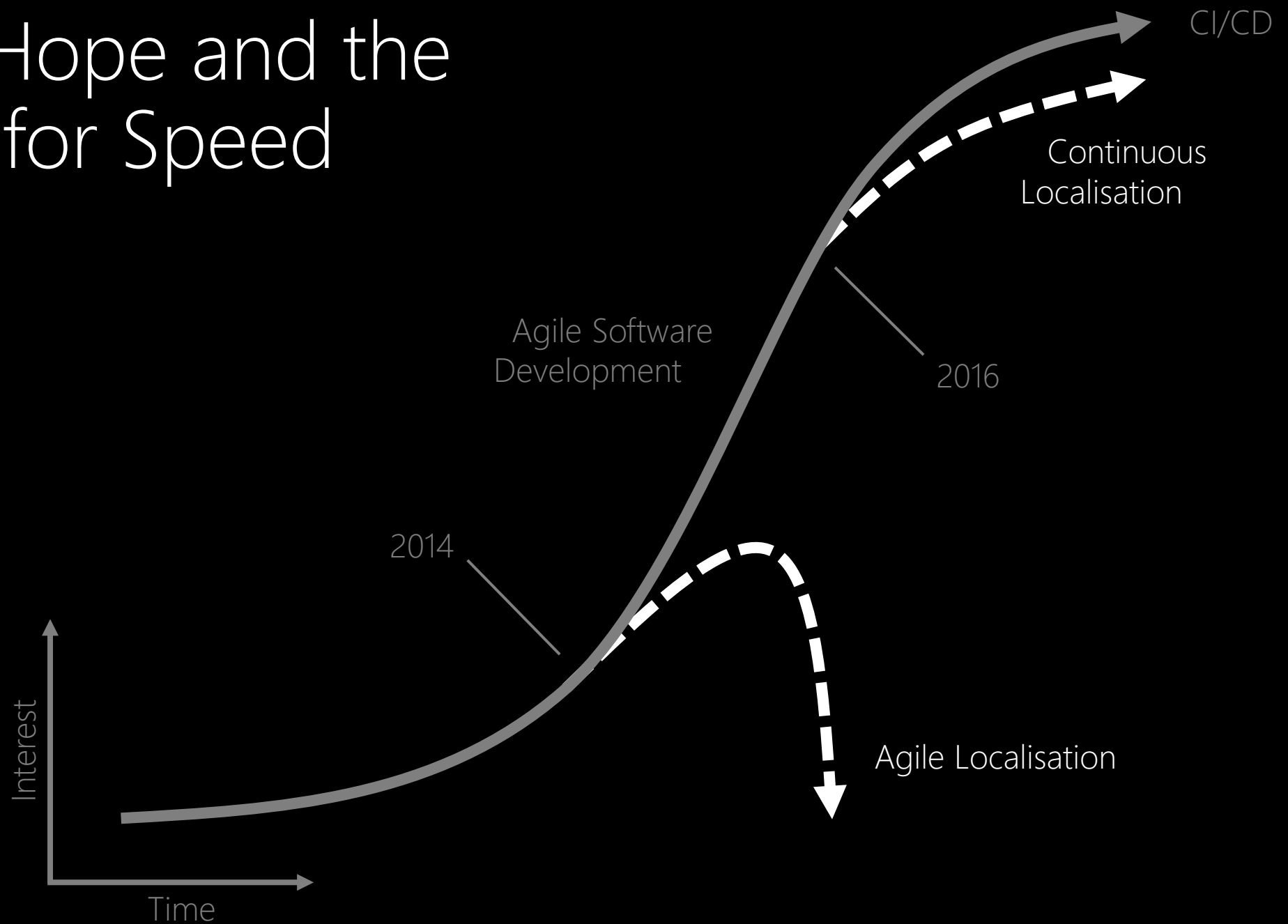
[linkedin.com/in/lefman](https://www.linkedin.com/in/lefman)



What you will learn

- Legacy
- Architecture
- Technology

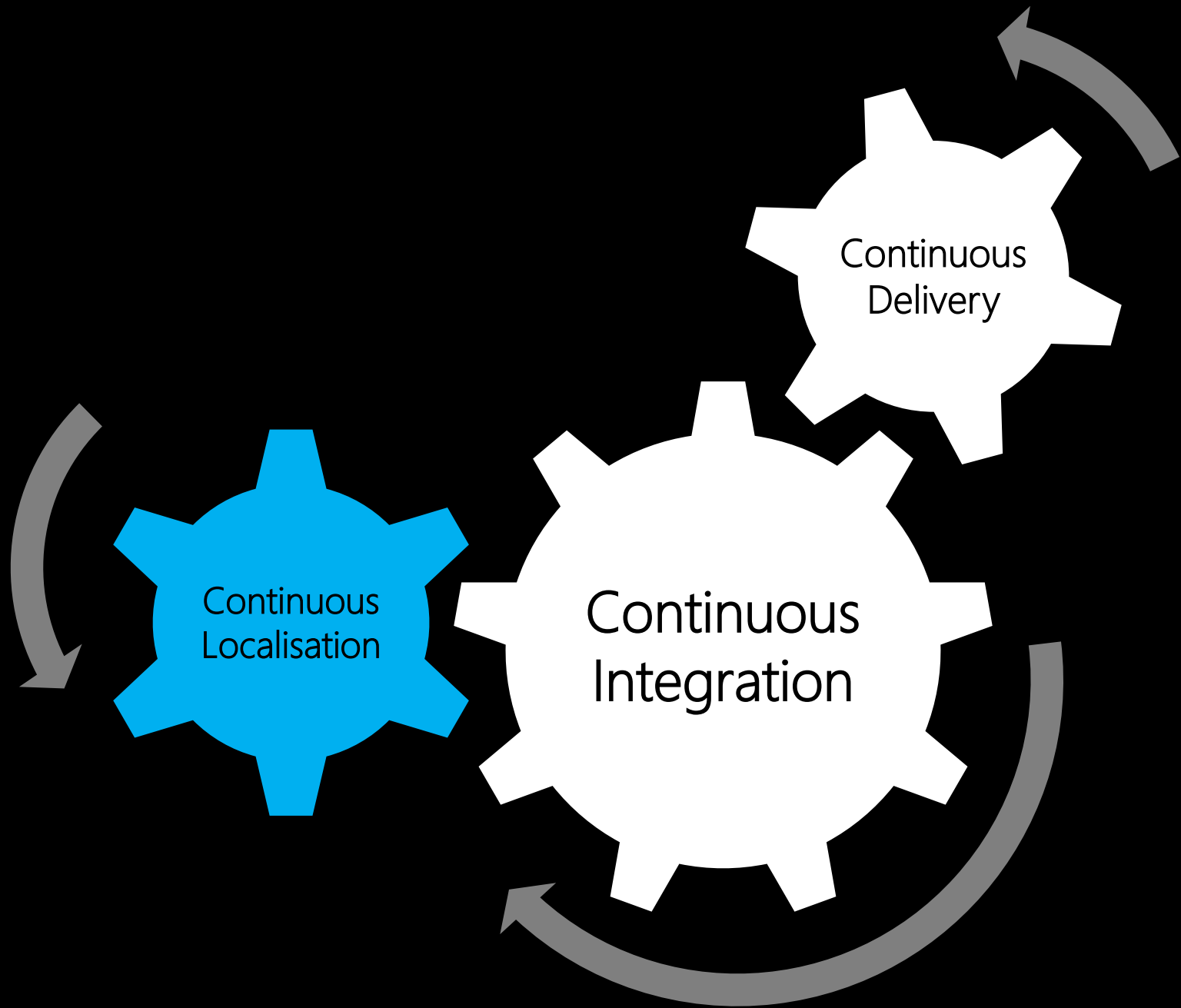
False Hope and the Need for Speed





“A collection of autonomous parts that form an uninterrupted process of localisation.”

Continuous Localisation



Continuous
Localisation

Continuous
Integration

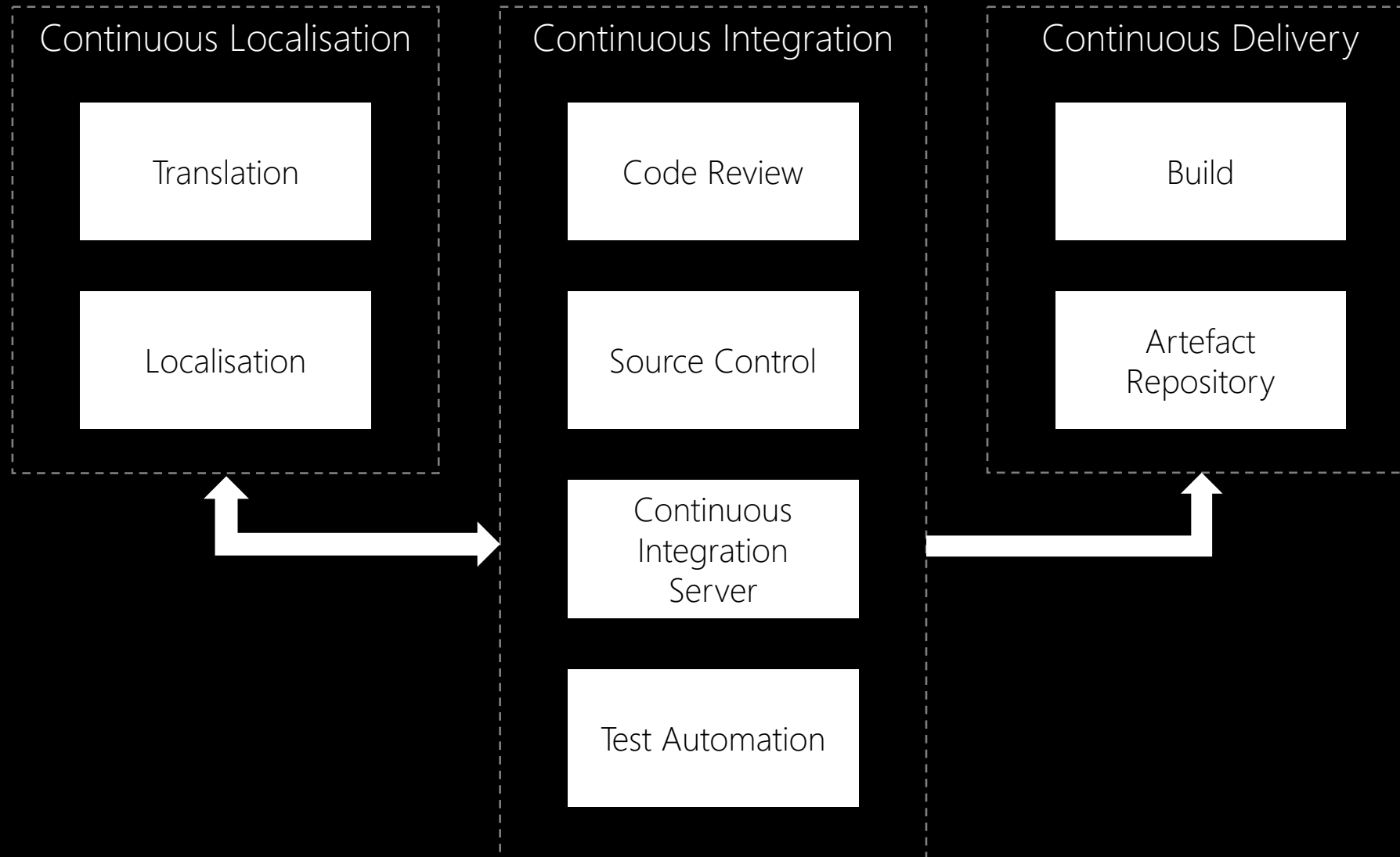
Continuous
Delivery

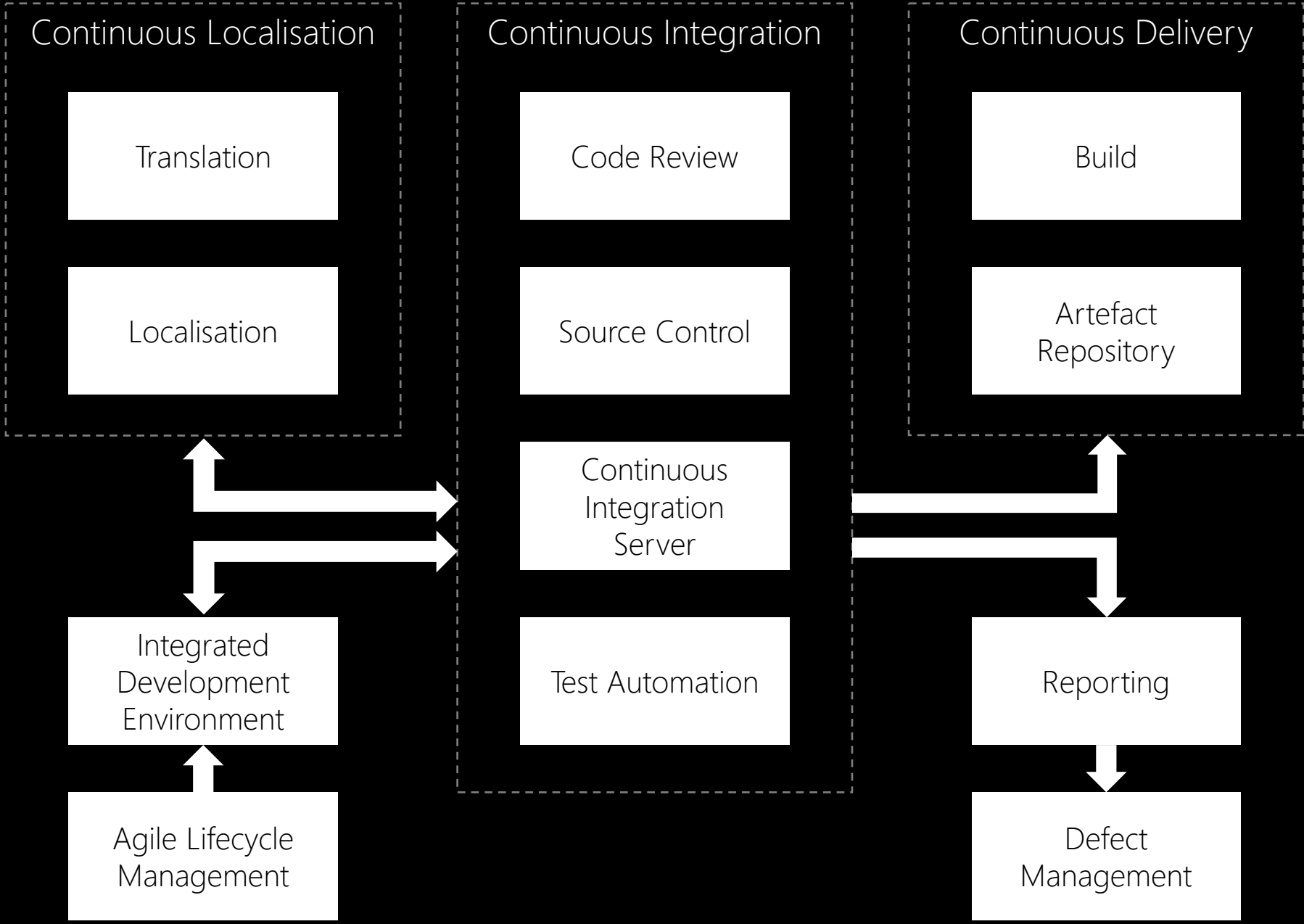


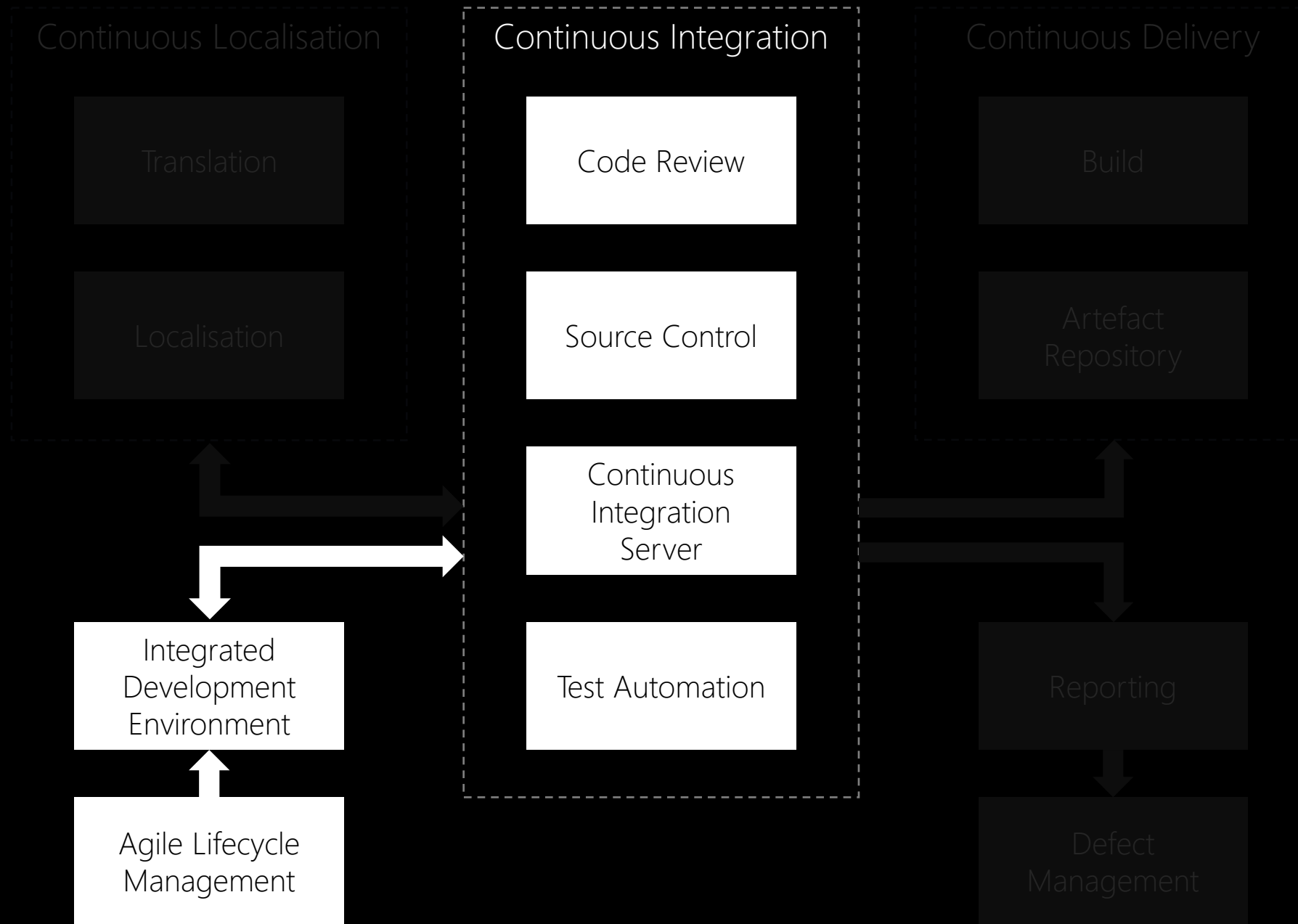
Continuous Localisation

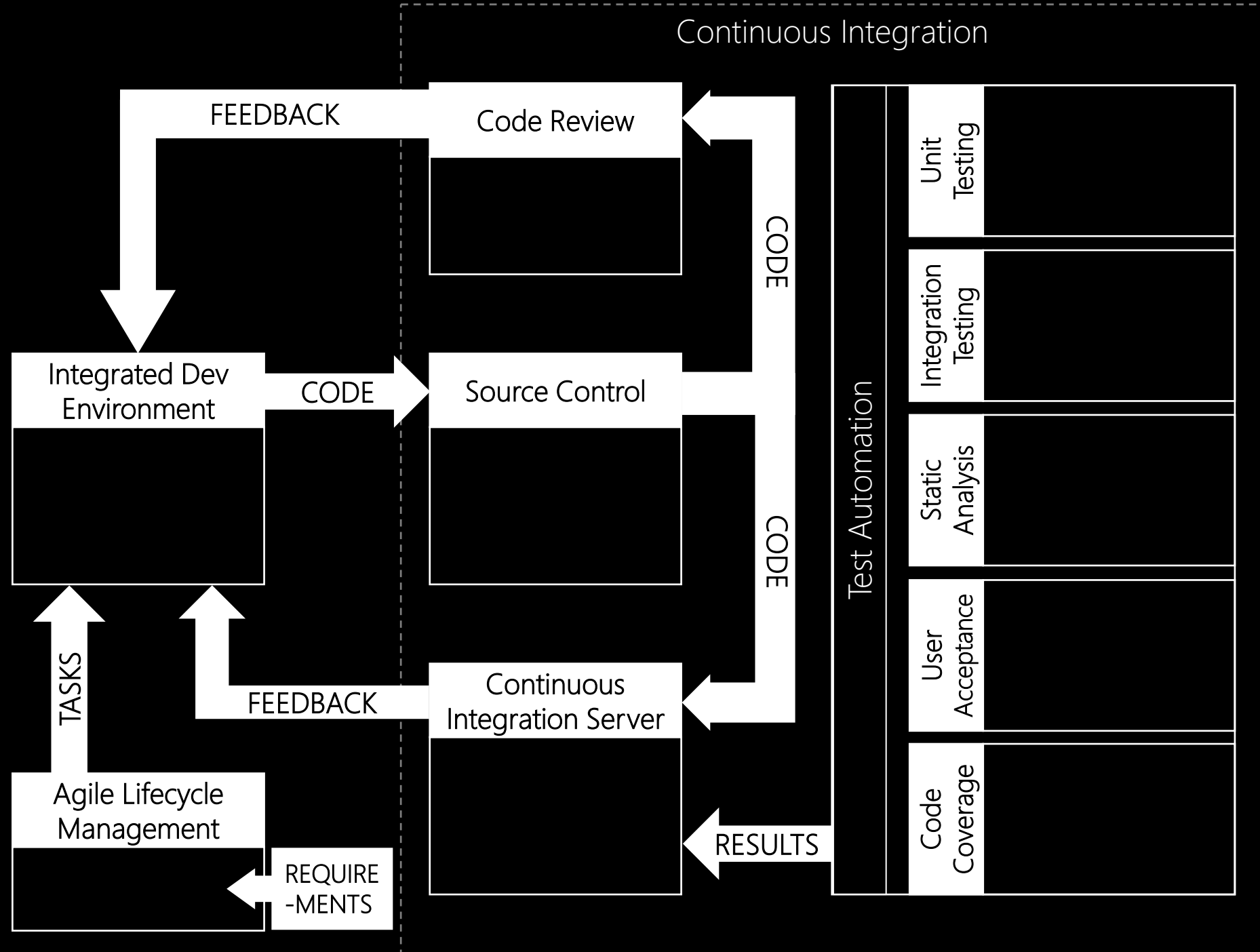
Continuous Integration

Continuous Delivery



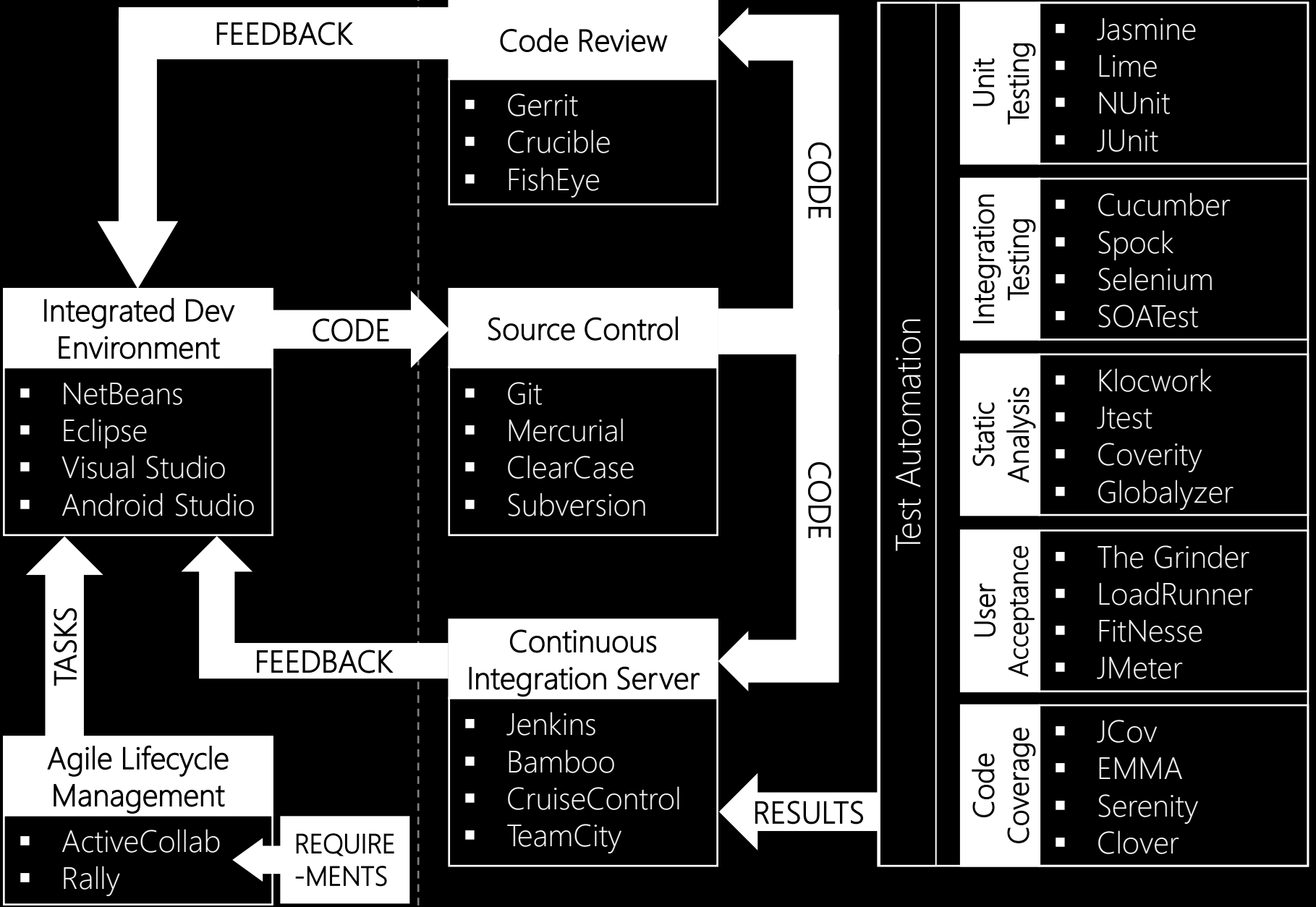








Continuous Integration



Integrated Dev Environment

- NetBeans
- Eclipse
- Visual Studio
- Android Studio

Code Review

- Gerrit
- Crucible
- FishEye

Source Control

- Git
- Mercurial
- ClearCase
- Subversion

Continuous Integration Server

- Jenkins
- Bamboo
- CruiseControl
- TeamCity

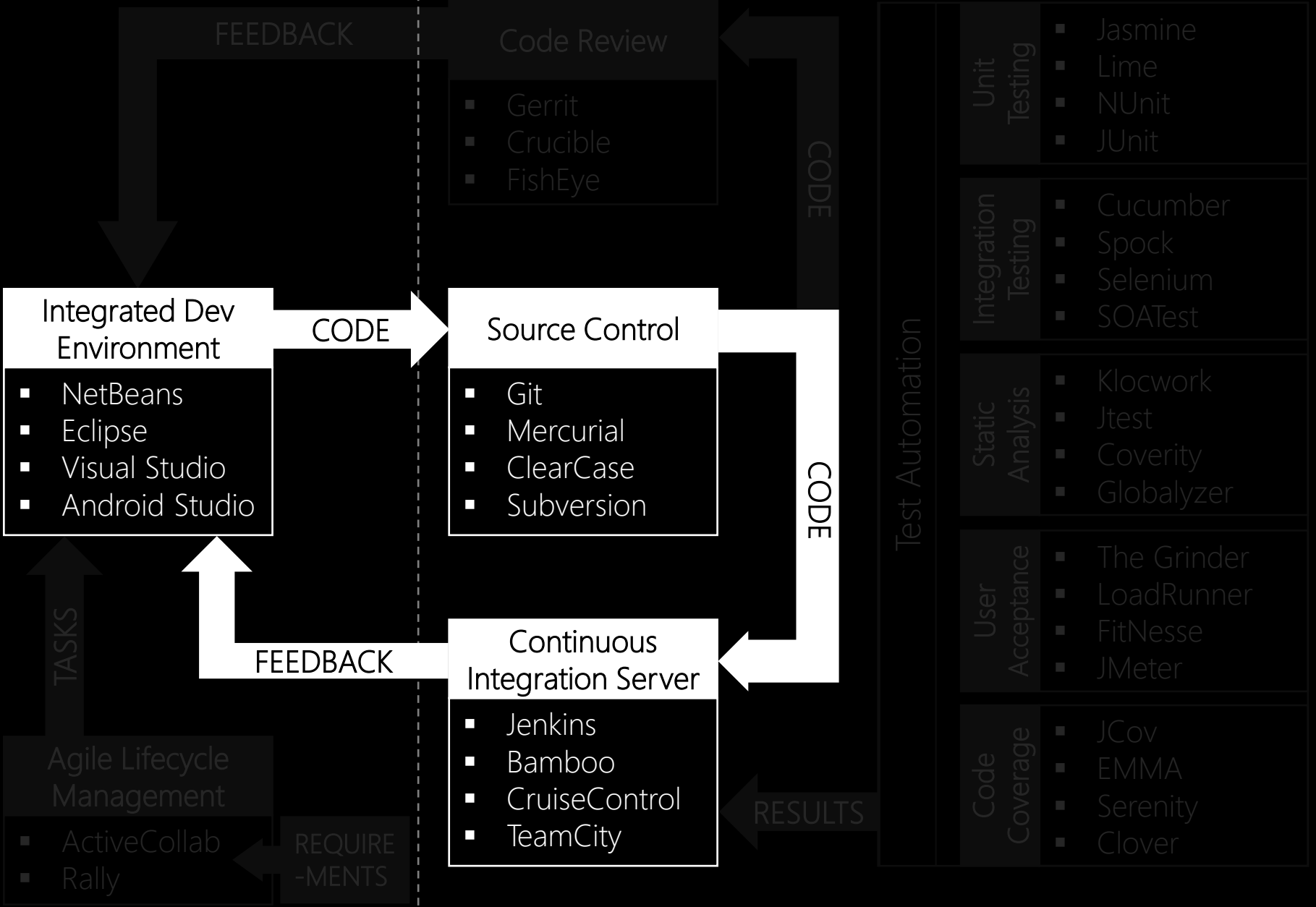
Agile Lifecycle Management

- ActiveCollab
- Rally

Test Automation	Unit Testing	<ul style="list-style-type: none"> ▪ Jasmine ▪ Lime ▪ NUnit ▪ JUnit
	Integration Testing	<ul style="list-style-type: none"> ▪ Cucumber ▪ Spock ▪ Selenium ▪ SOATest
	Static Analysis	<ul style="list-style-type: none"> ▪ Klocwork ▪ Jtest ▪ Coverity ▪ Globalyzer
	User Acceptance	<ul style="list-style-type: none"> ▪ The Grinder ▪ LoadRunner ▪ FitNesse ▪ JMeter
	Code Coverage	<ul style="list-style-type: none"> ▪ JCov ▪ EMMA ▪ Serenity ▪ Clover



Continuous Integration



Agile Lifecycle Management

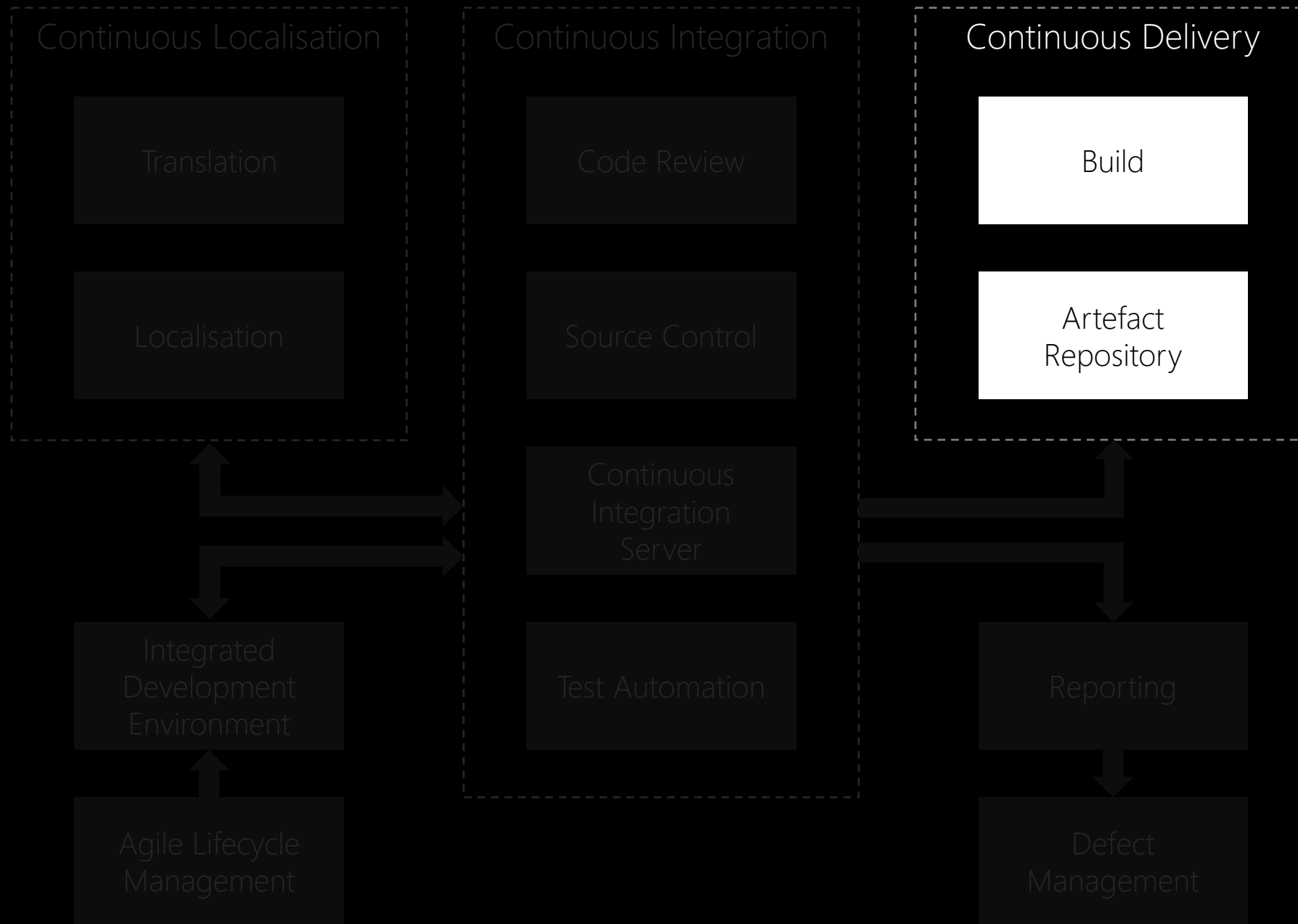
- ActiveCollab
- Rally

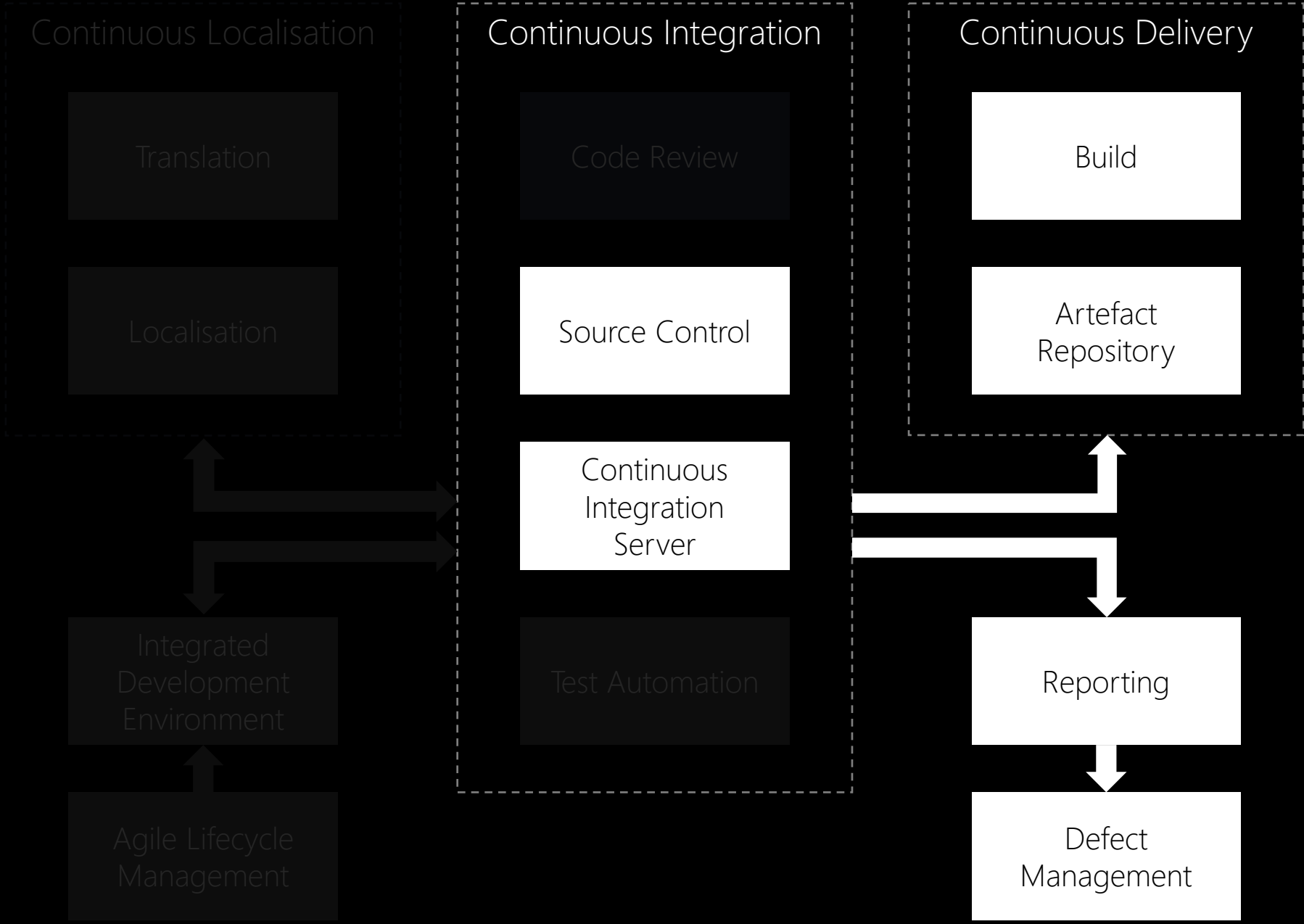
REQUIREMENTS

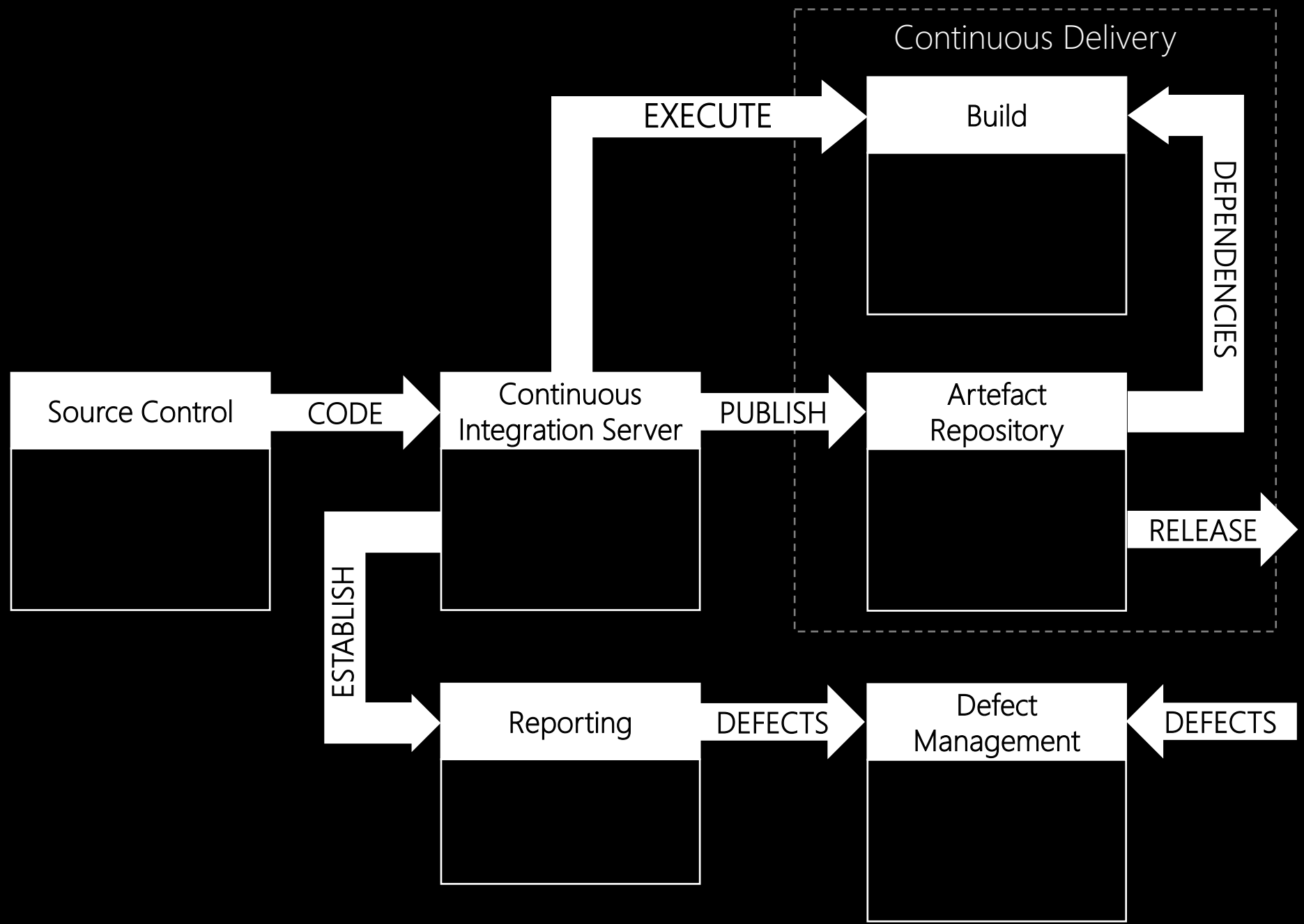
Code Review

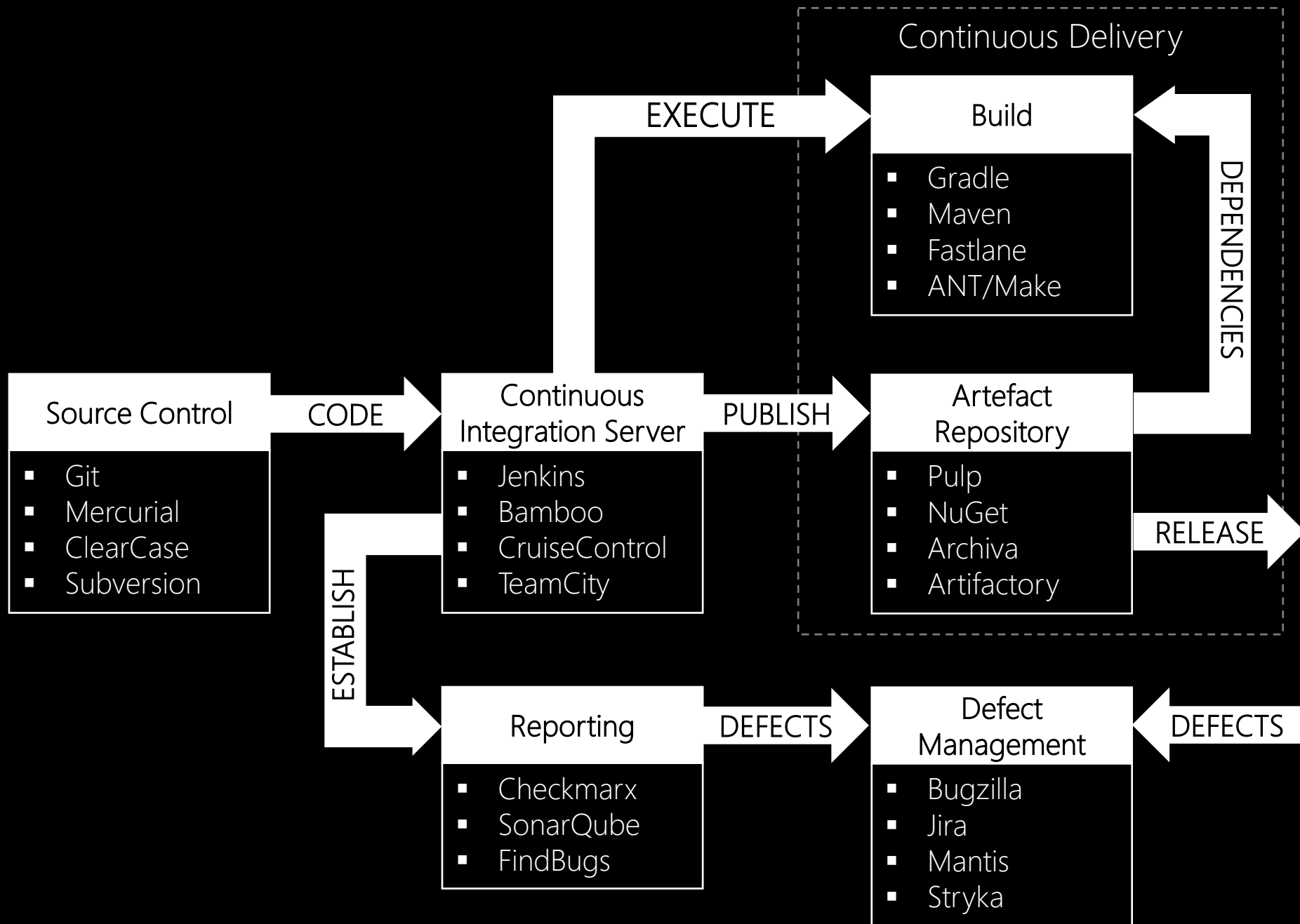
- Gerrit
- Crucible
- FishEye

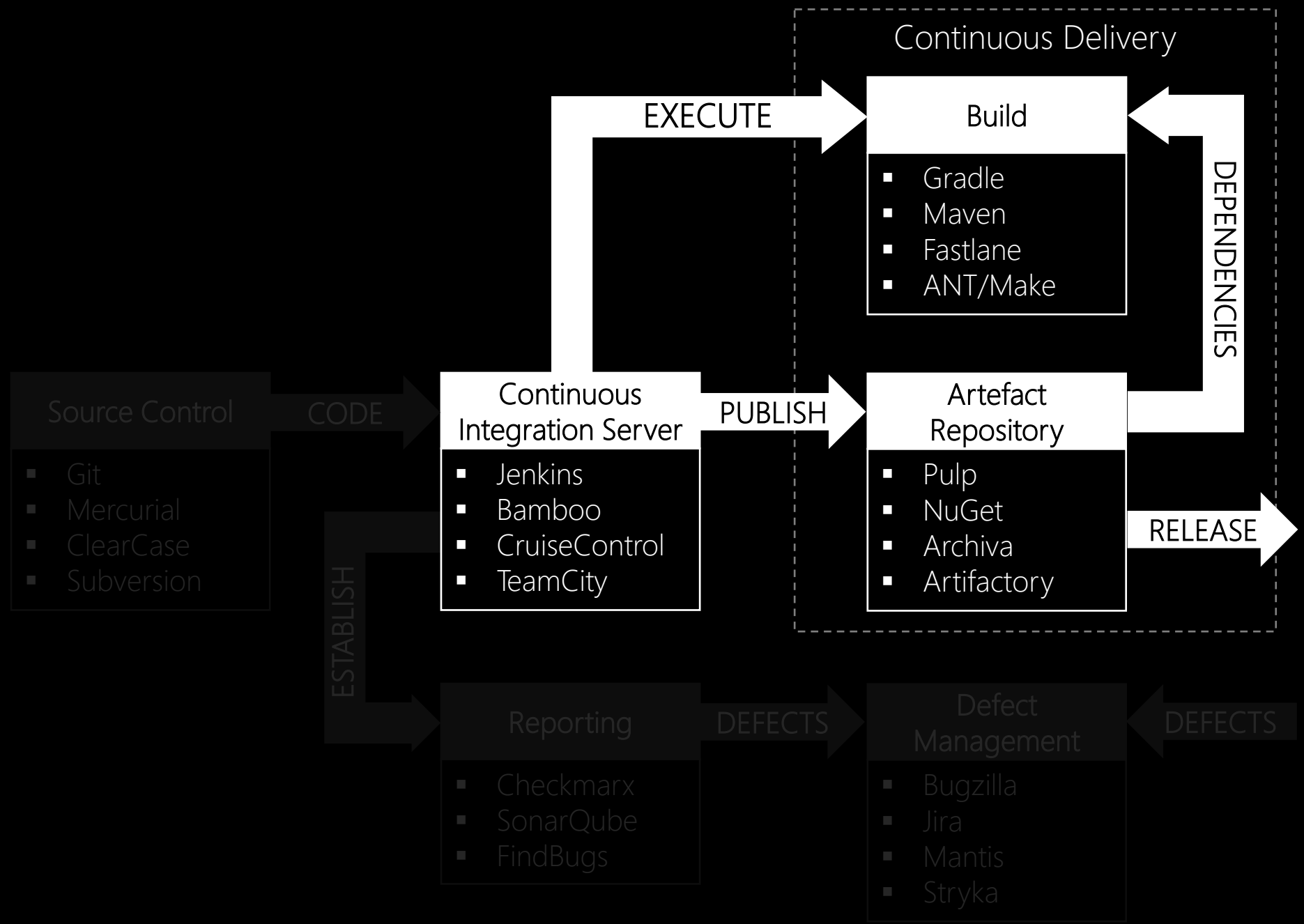
Test Automation	Unit Testing	<ul style="list-style-type: none">JasmineLimeNUnitJUnit
	Integration Testing	<ul style="list-style-type: none">CucumberSpockSeleniumSOATest
	Static Analysis	<ul style="list-style-type: none">KlocworkJtestCoverityGlobalyzer
	User Acceptance	<ul style="list-style-type: none">The GrinderLoadRunnerFitNesseJMeter
	Code Coverage	<ul style="list-style-type: none">JcovEMMASerenityClover





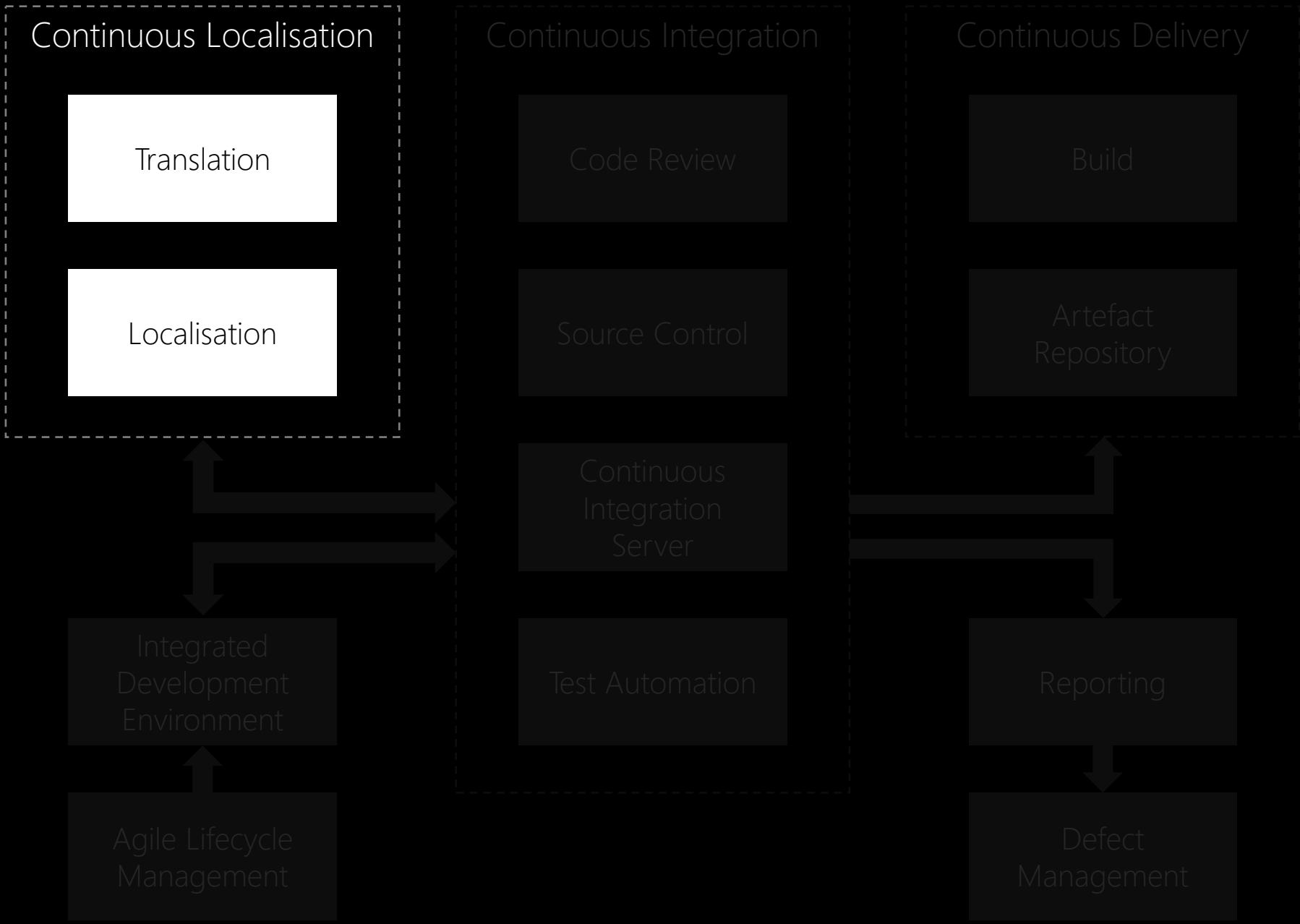


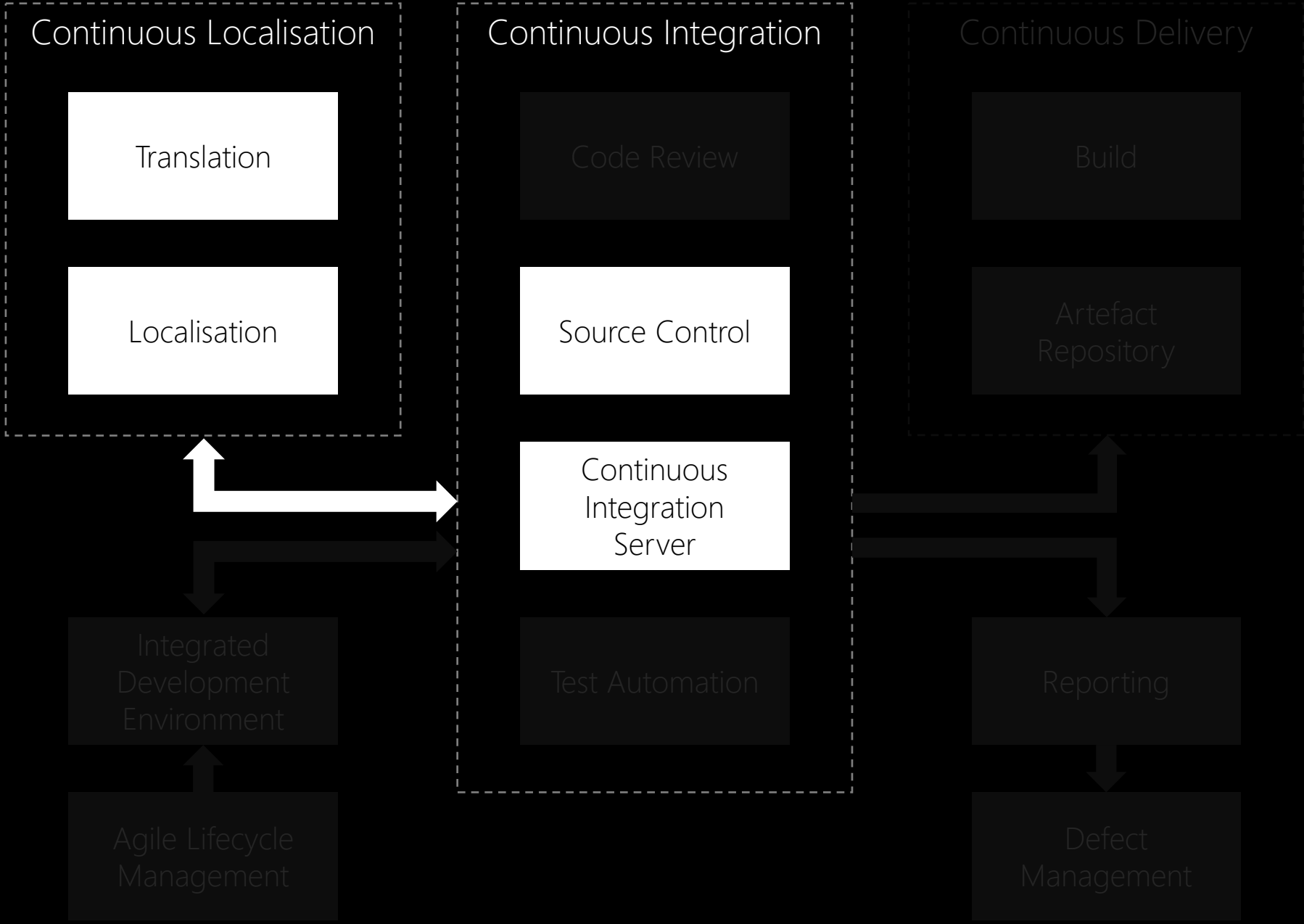


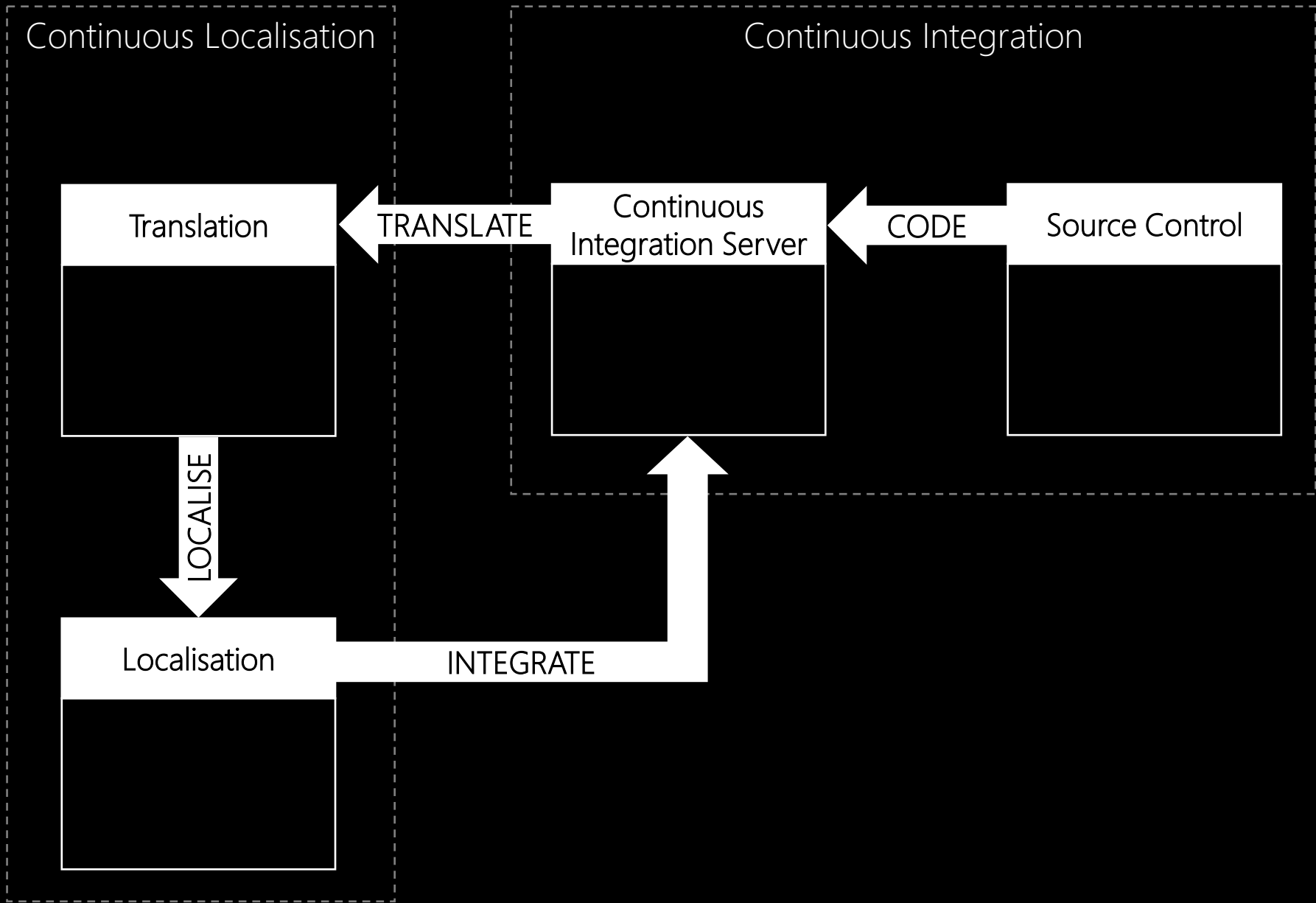


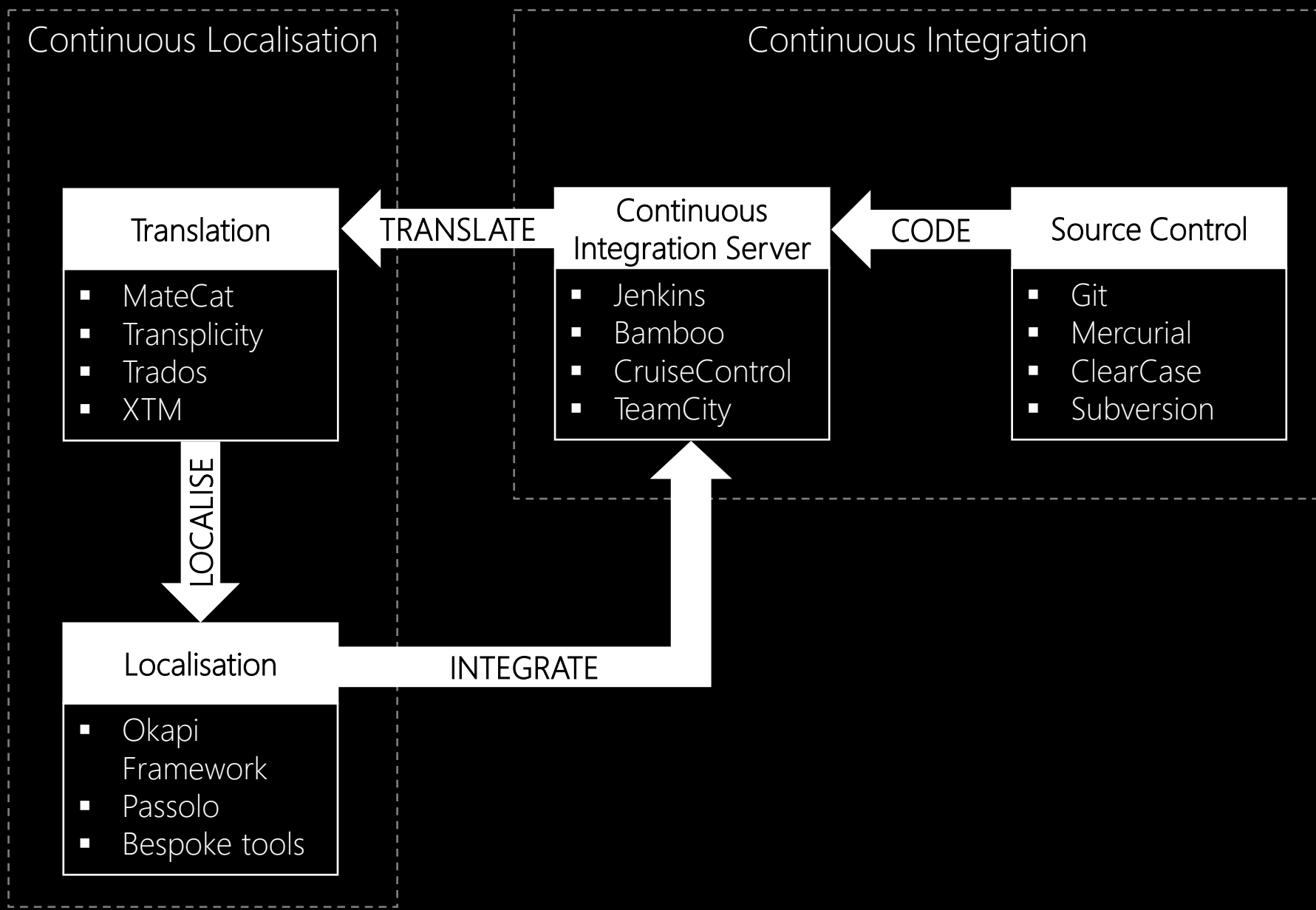


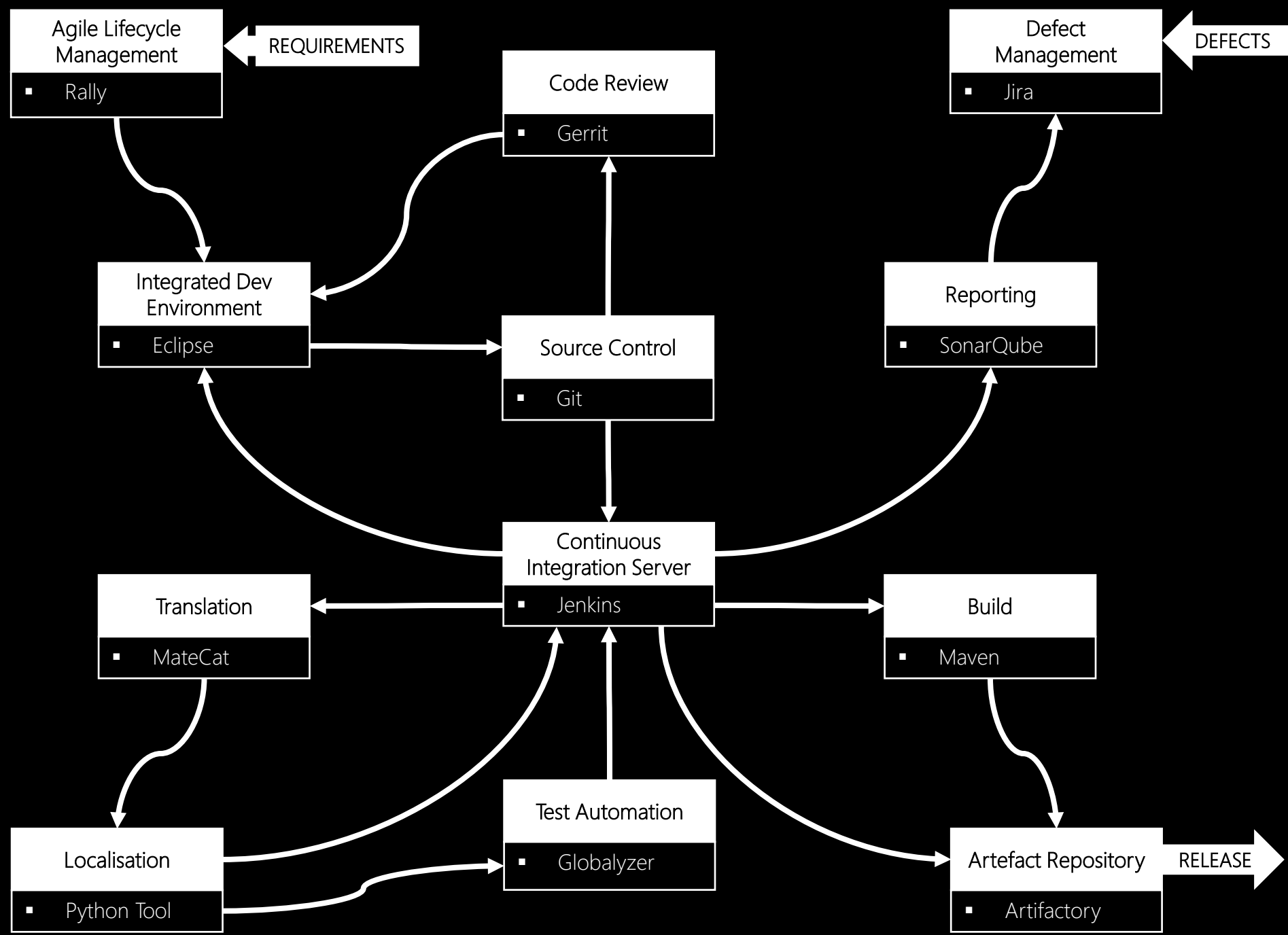
Reduce the **distance** between
development and localisation.













How can this be achieved





A change of *attitude*.

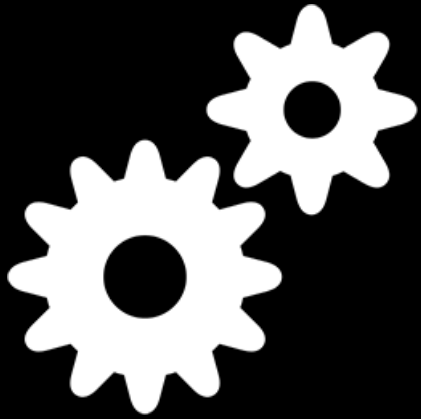
Throw away



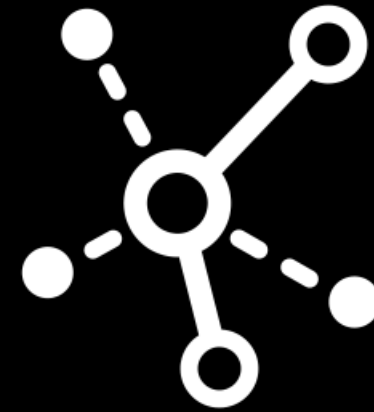
traditional processes.



Continuous localisation
cannot be achieved
without automation.

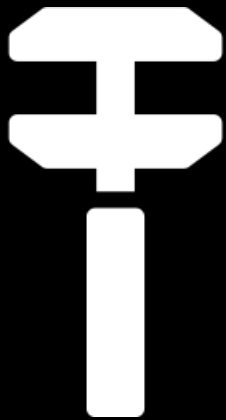


Tools and
Services

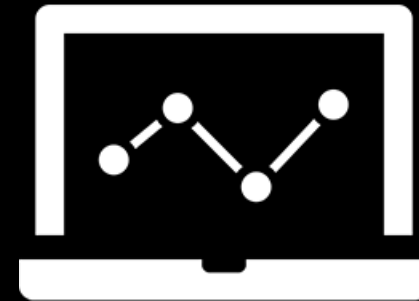


Well-defined
Interfaces

Automation



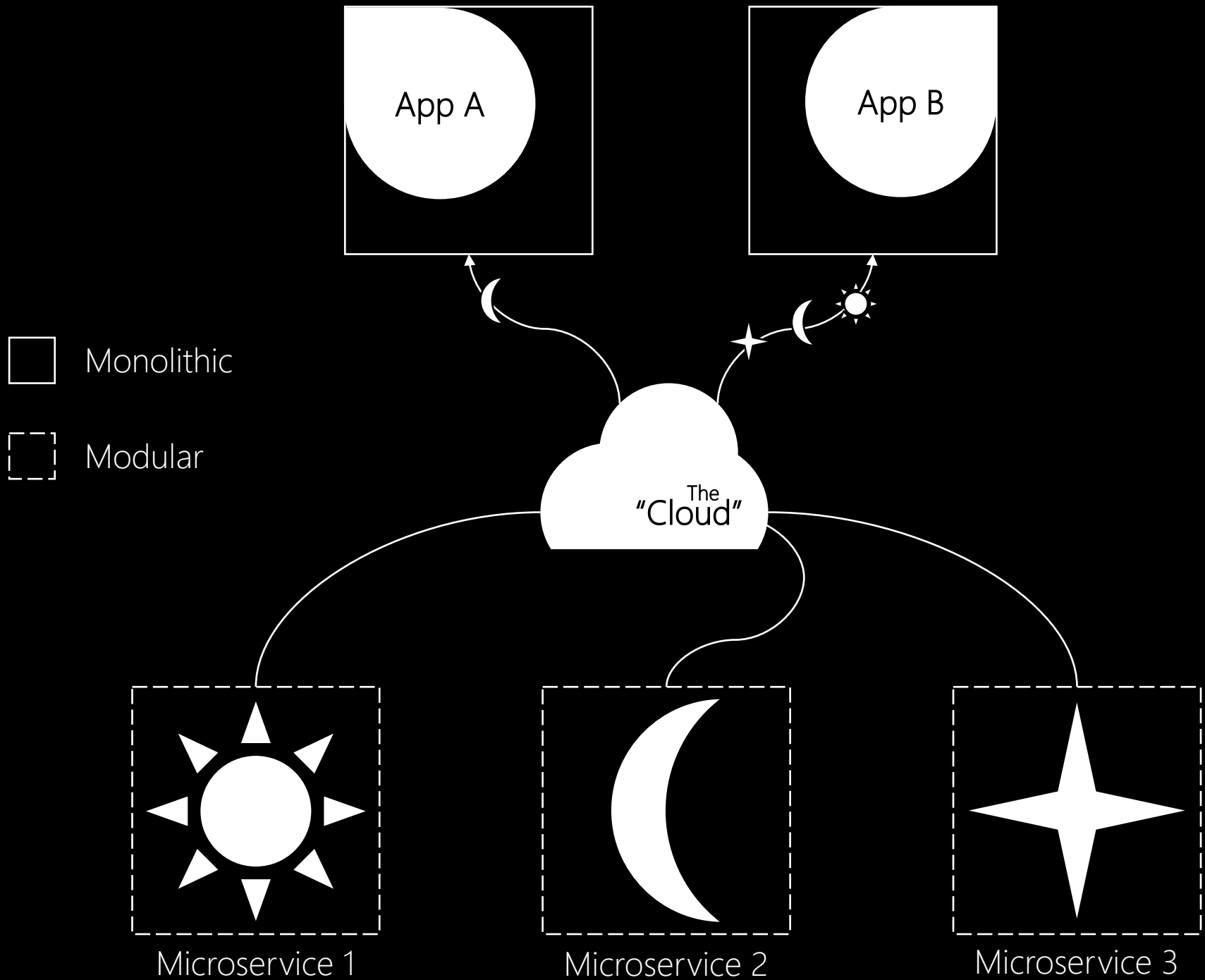
Maintenance and
Enhancement

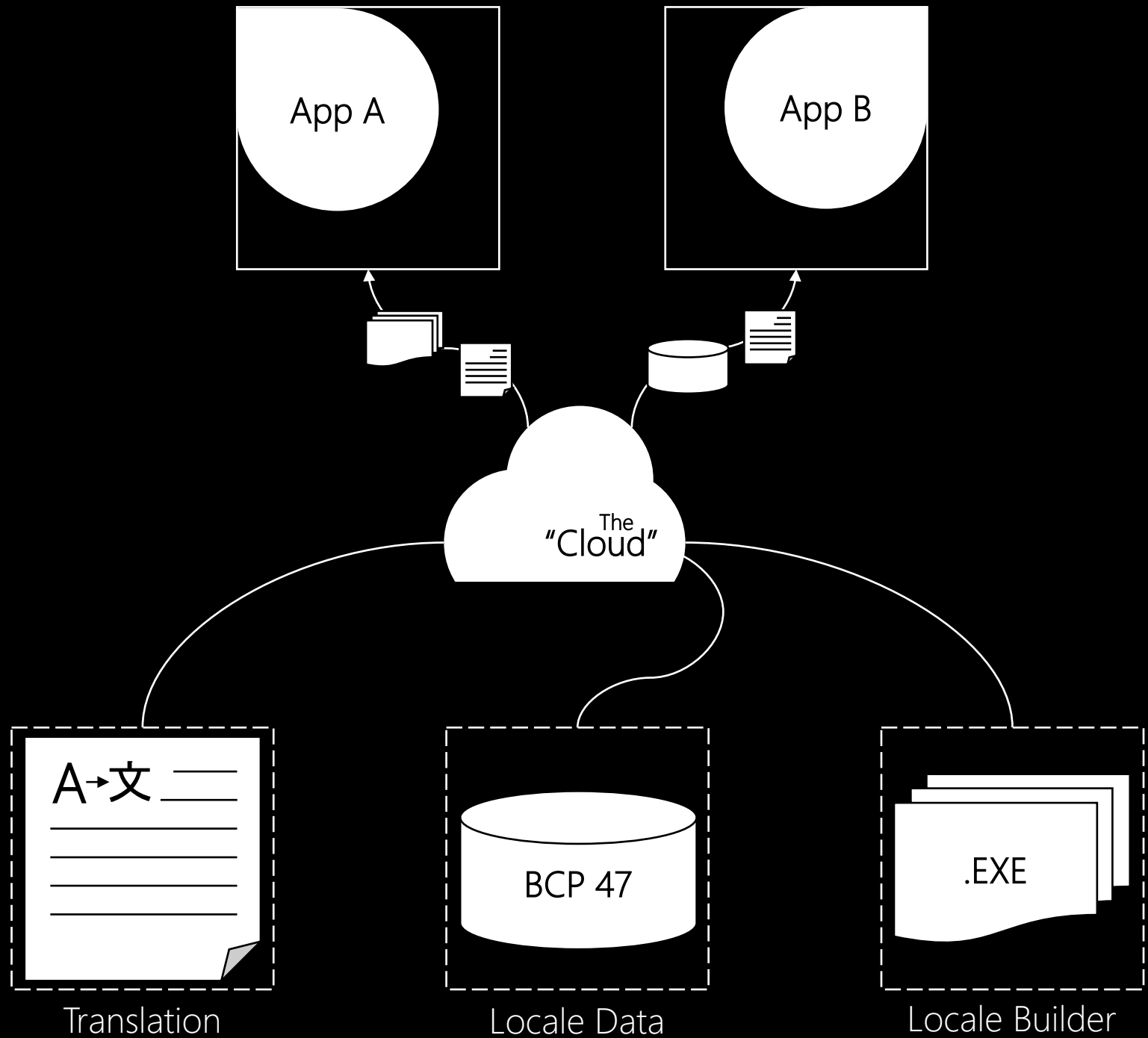


Monitoring
and Reporting



Integration between tools and services are the **arteries** of continuous localisation.

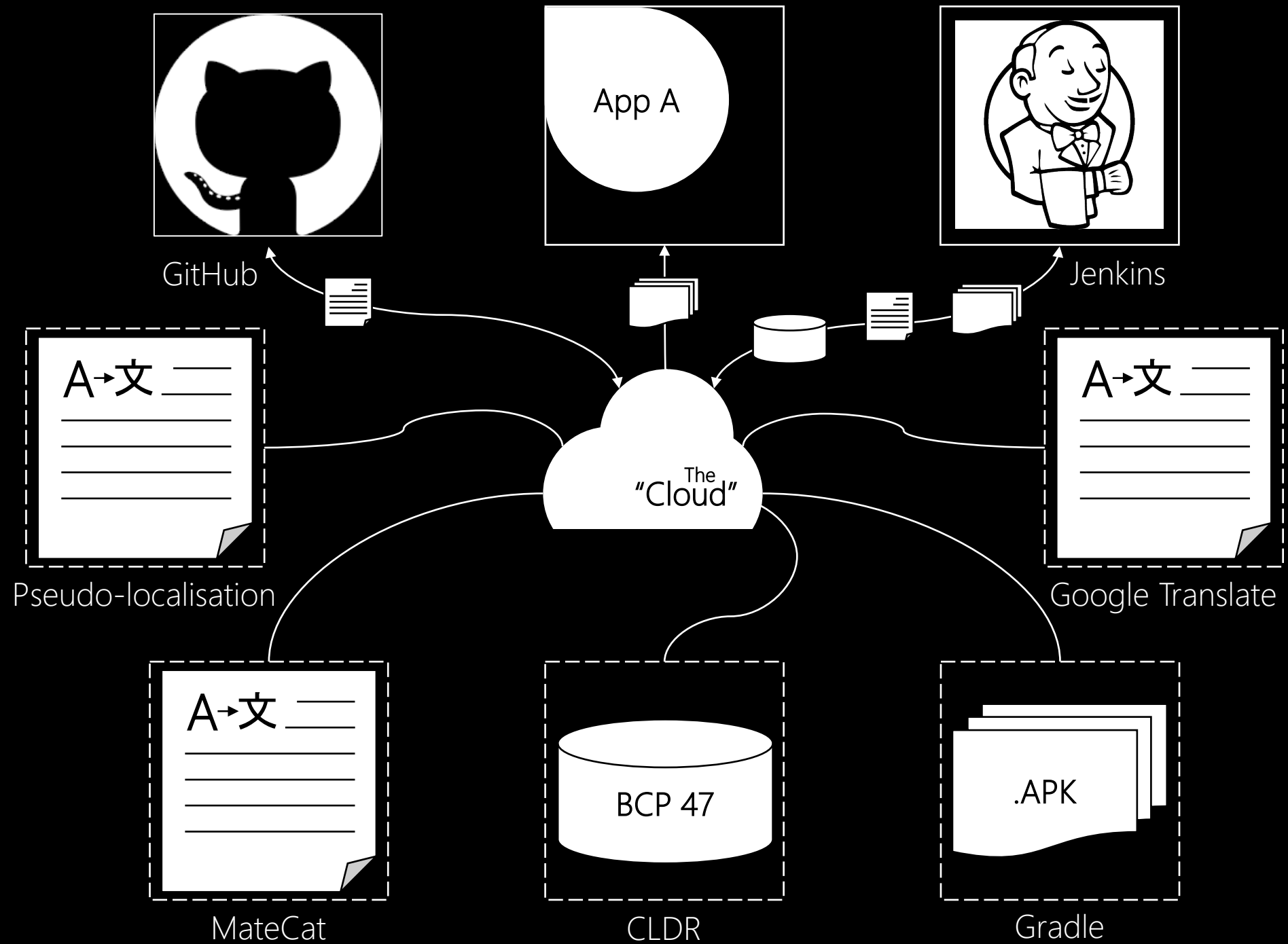


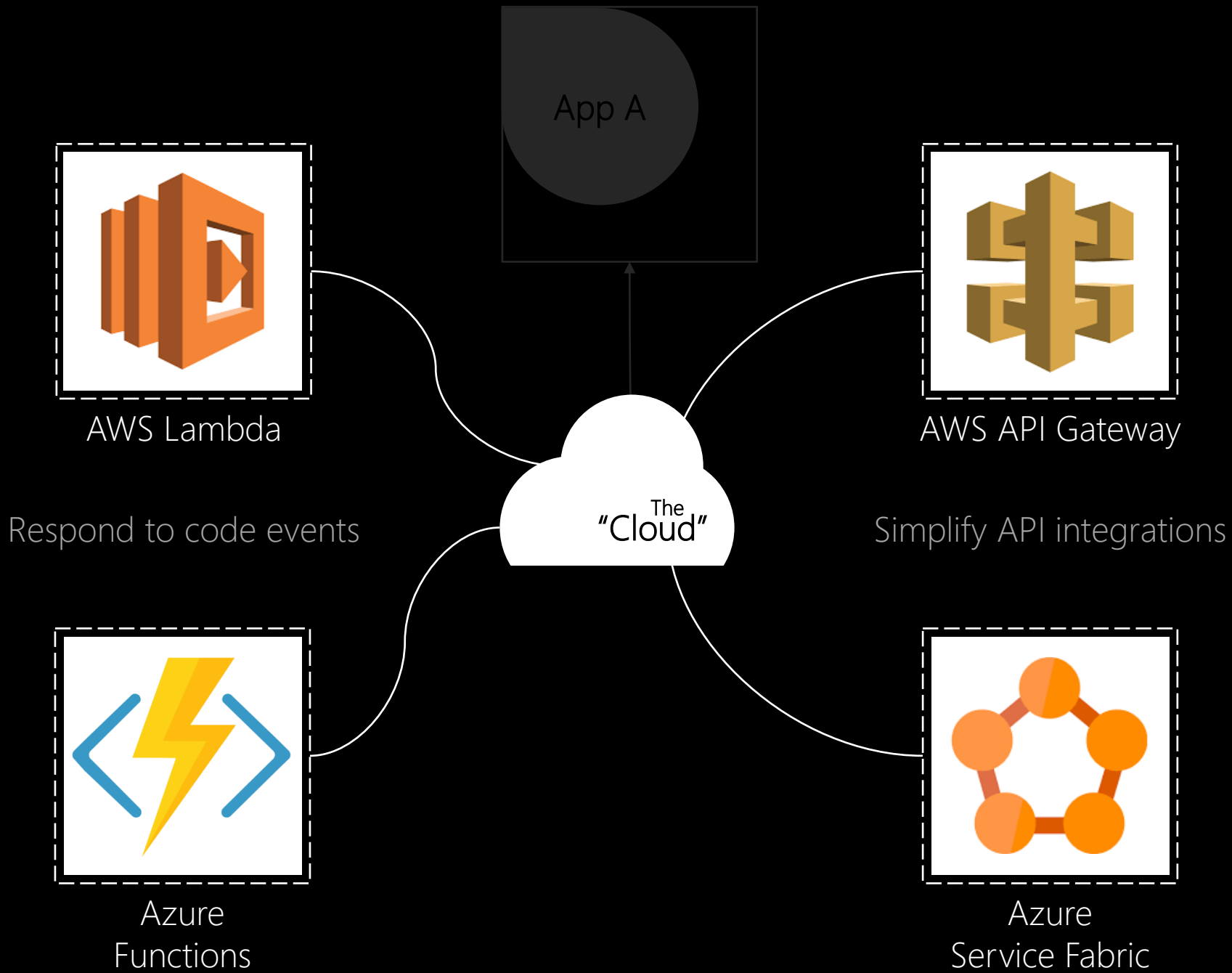


Translation

Locale Data

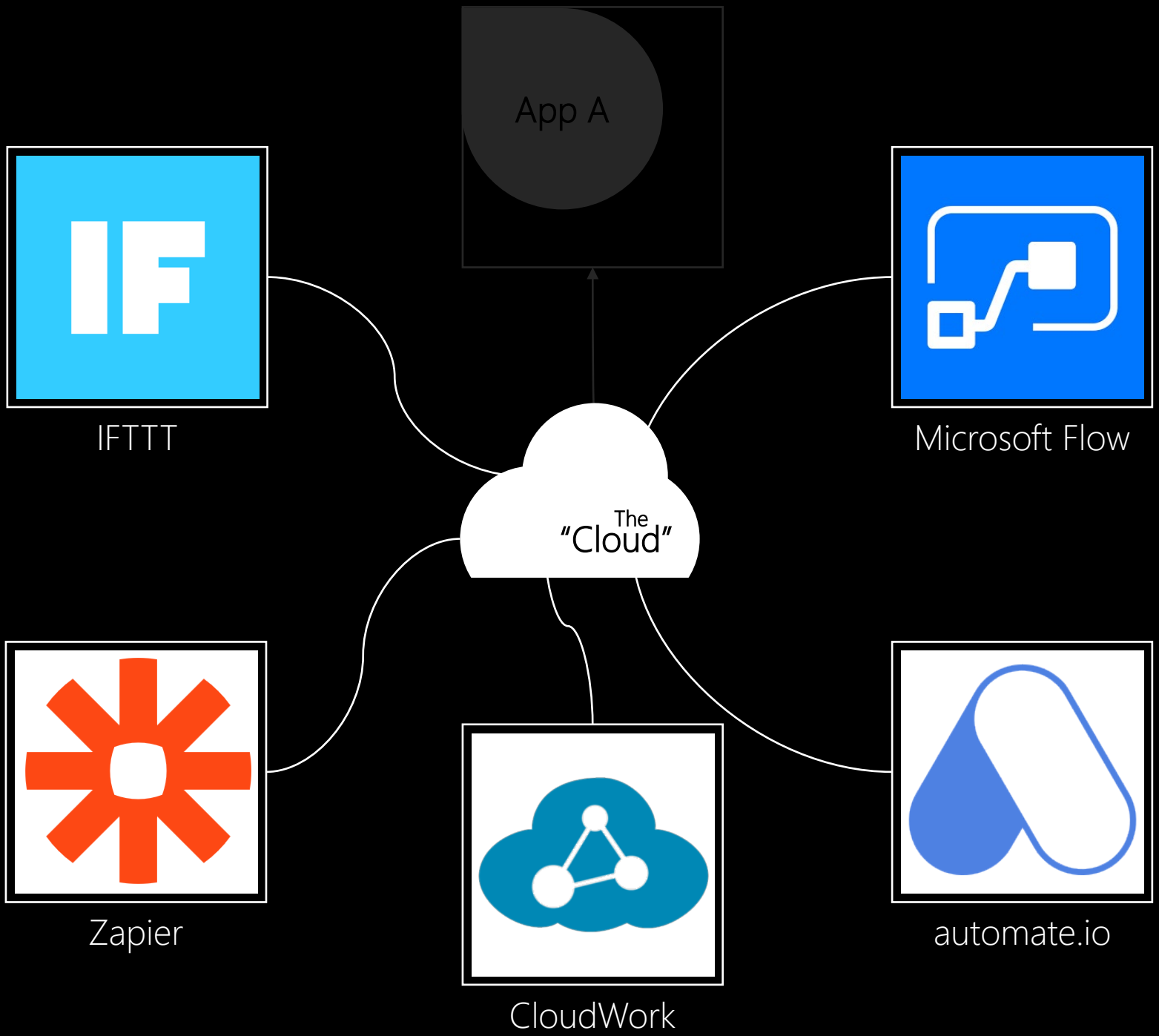
Locale Builder



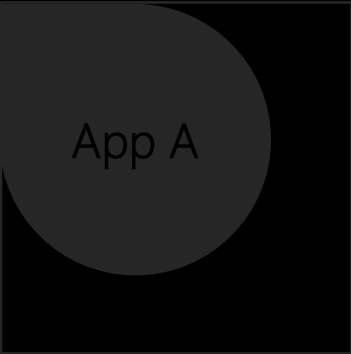




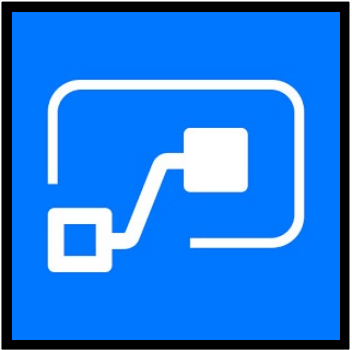
Why stop here?



IFTTT



App A



Microsoft Flow



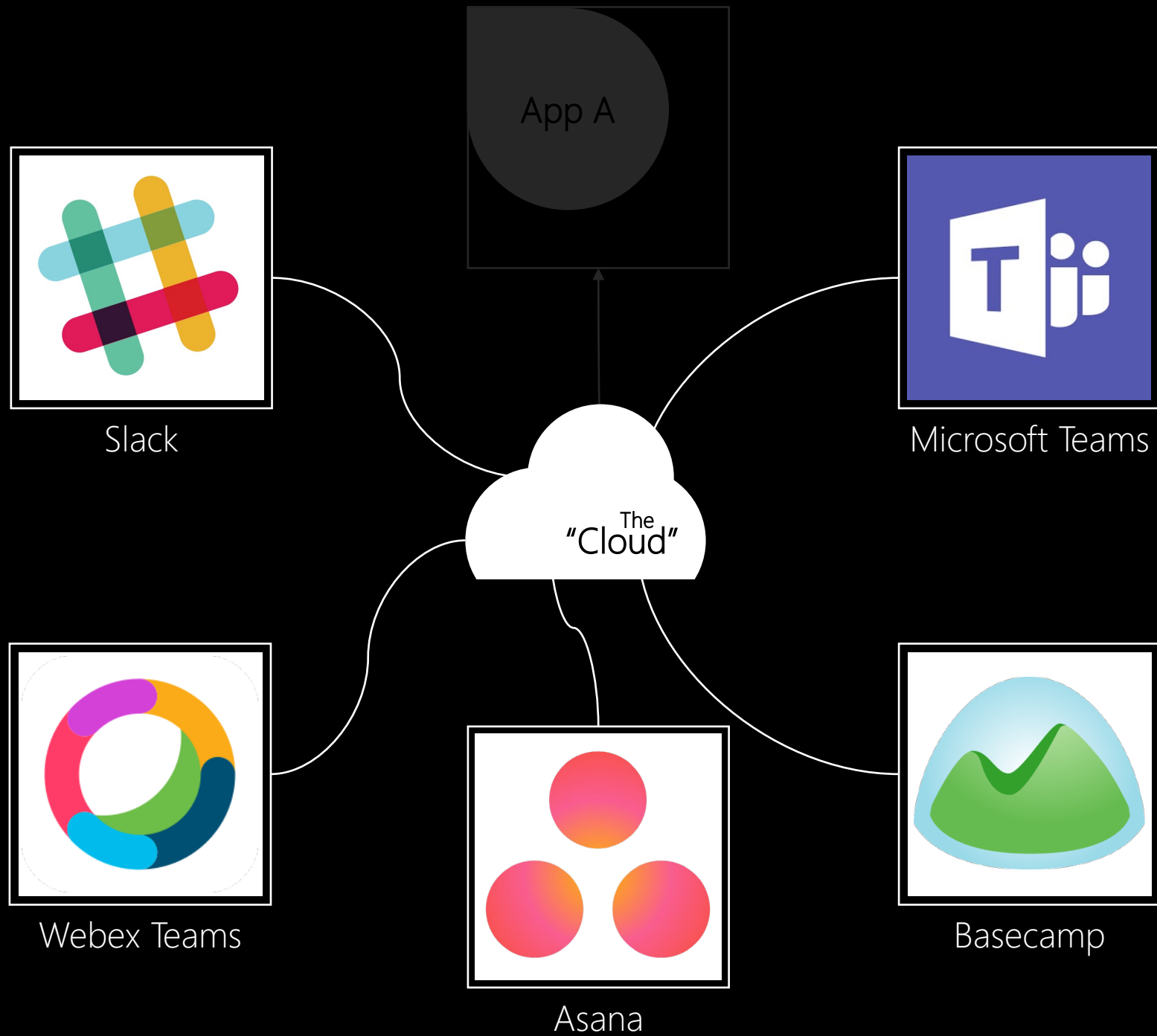
Zapier

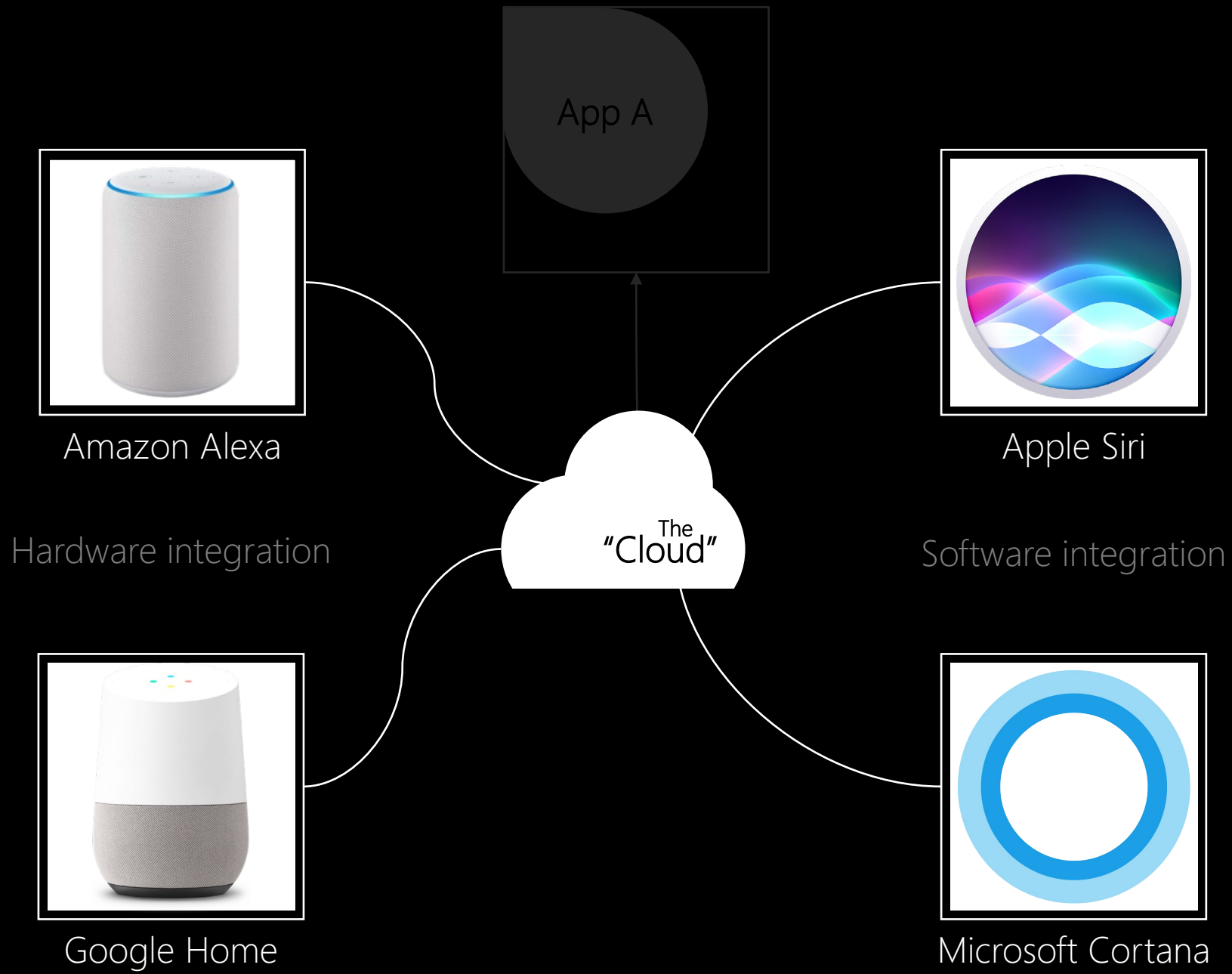


CloudWork



automate.io







Your takeaways

- Legacy
- Architecture
- Technology

Questions?



[linkedin.com/in/lefman](https://www.linkedin.com/in/lefman)

Session Abstract

Throughout 2016, attendees of the localisation industry's leading conference were talking feverishly about **continuous localisation** as if it were a new fashionable term for the localisation industry. However, when asked how it was working out for them, the response was frequently along the lines of "we still have a long way to go" or the less encouraging "we haven't even started yet".

This brief obsession with the **agile software development methodology** amounted to nothing significant, but it did fuel the desire for **faster localisation**. To make this possible, there needed to be a **seismic shift** in the way software is **developed and released**.

Continuous localisation is software **internationalisation** and **localisation** processes fused with a **continuous integration and deployment** environment. This session is for people in organisations that are thinking about continuous localisation. We look at the ingredients for **blending localisation processes with a new or existing continuous integration environment**. We also examine the **tools and interfaces** that can be used to **automate continuous deployment of locales**.

By the end of the session you will recognise the key elements needed to **design** a continuous localisation **platform**, and you will understand the marvels of **modern gadgets and gizmos** that make it much more than just an agile talking point.