CONNECTING THE ECLIPSE IDE TO THE CLOUD-BASED ERA OF DEVELOPER TOOLING

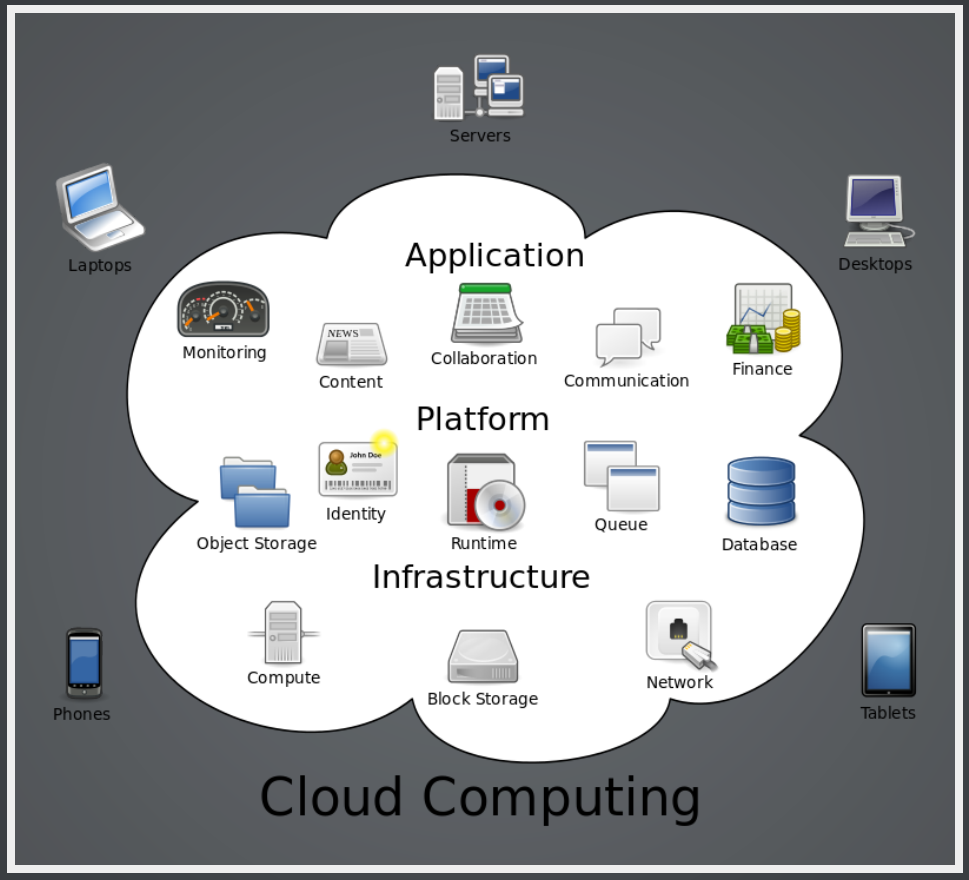
JAVAONE 2014

Martin Lippert - Pivotal / @martinlippert
John Arthorne - IBM / @jarthorne

'I CAN ONLY EVER CODE ON ONE LAPTOP BECAUSE IT IS SO #$@&%*! HARD TO SETUP MY TOOLS THE WAY I WANT THEM'

'I NEED TO KNOW THE EXACT SET OF TOOLS USED TO BUILD THE SOFTWARE I DELIVERED THREE YEARS AGO.'

CLOUD AS TOOL PLATFORM



Source: Sam Johnston Creative Commons BY-SA 3.0

CLOUD ADVANTAGES

- Truly cross platform (tablets, servers, desktops ...)
- Extreme scalability (up and down)
- Enables live remote collaboration
- Low cost tool configuration
- Toolchain control

CURRENT STATE OF CLOUD TOOLING

- Current tools demonstrate what is possible
- Huge gaps in the available tool chain
- Will take years to catch up to desktop tools ecosystem

A DIFFICULT CHOICE

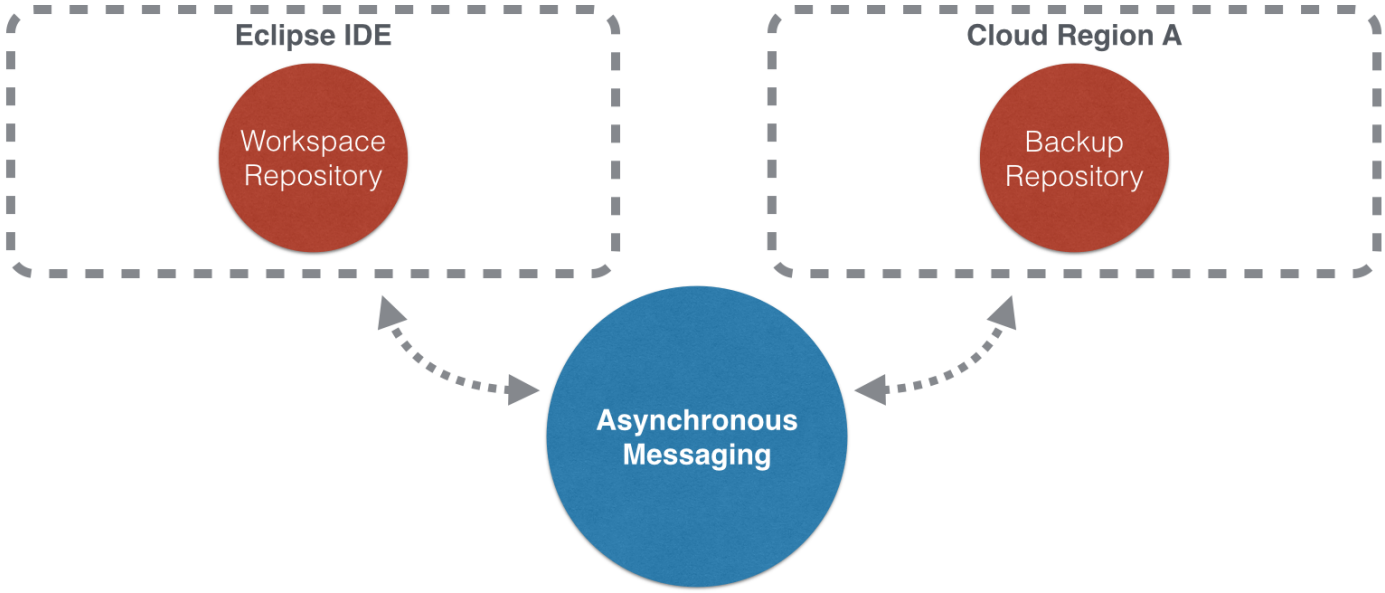
- Stay with desktop tools and live with desktop constraints
- Take the leap to the cloud, and leave existing tools behind
- But what if you could have both?

FLUX DEMO

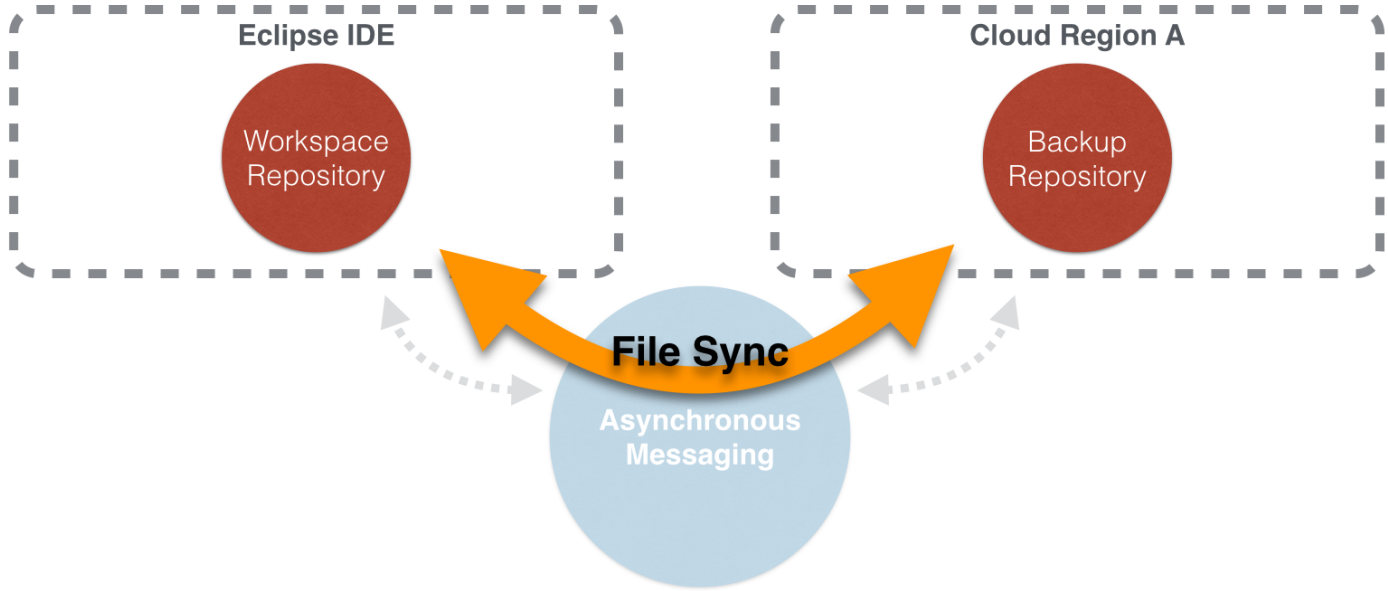
FLUX DEMO - THE DETAILS

- Connect Eclipse project to Flux
- Java syntax validation
- Content assist
- Jump to declaration (F3)
- Errors across editors
- Live Sync
- Browser only

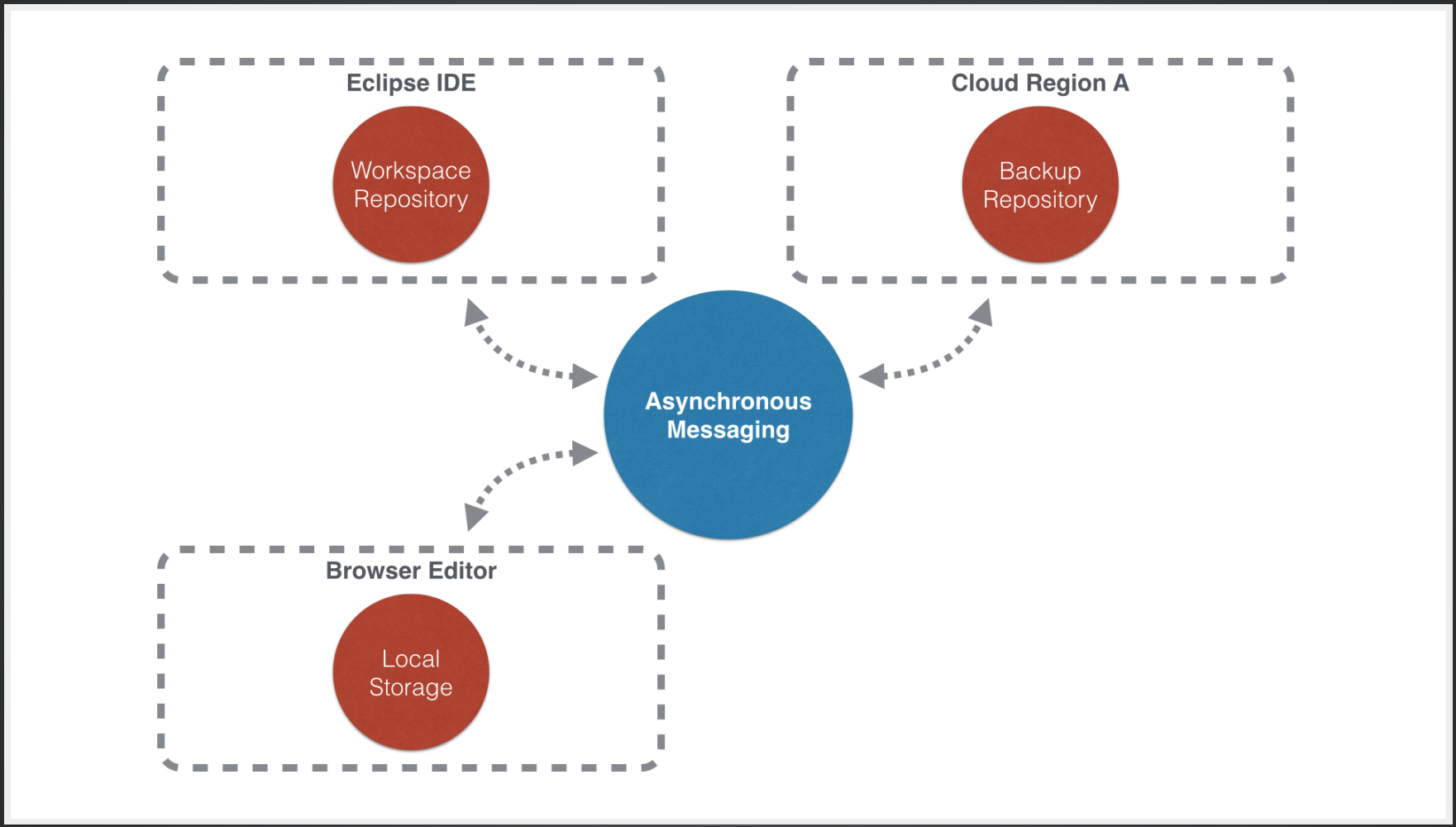
CURRENT ARCHITECTURE - SYNCING



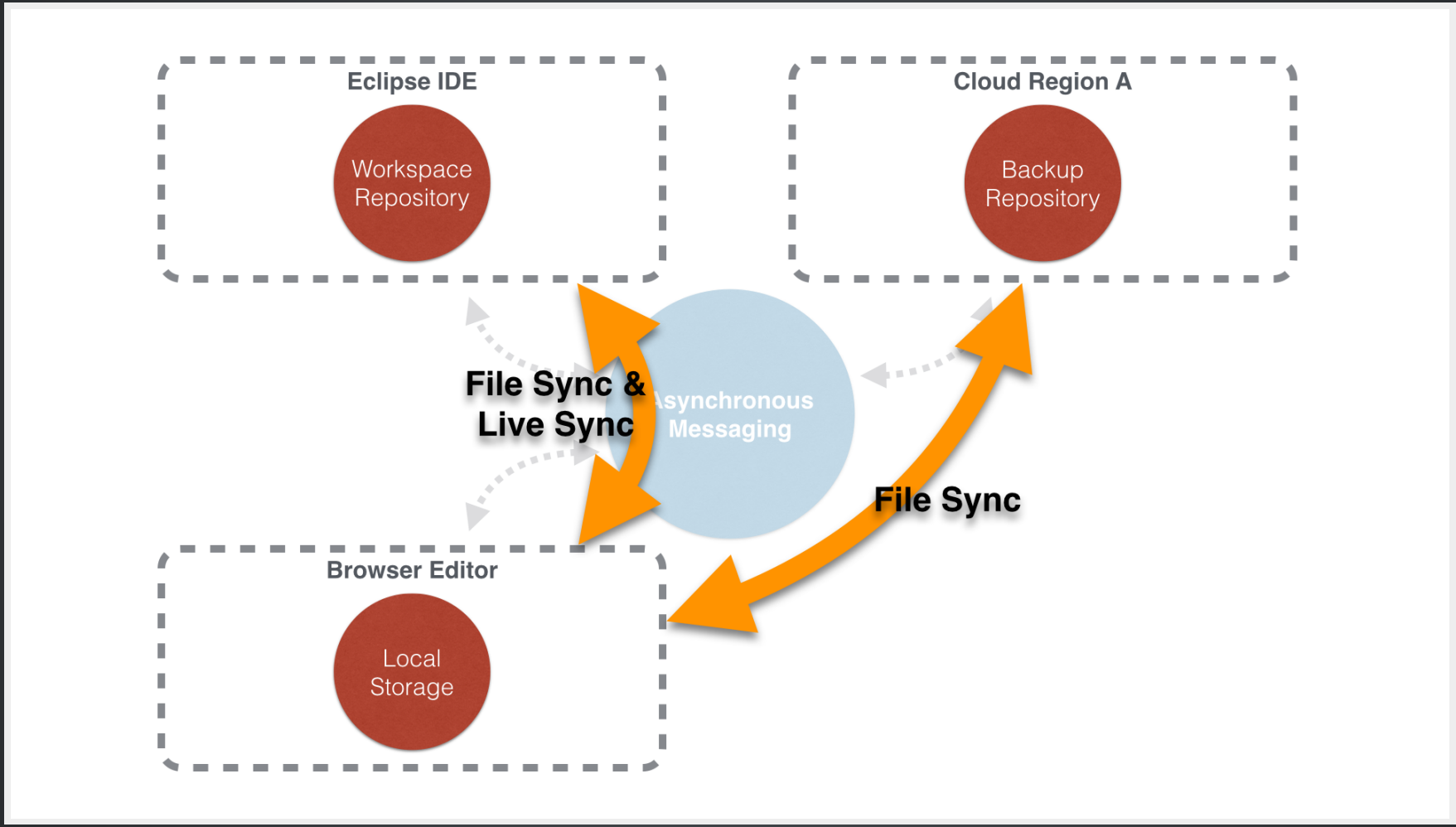
SYNCING RESOURCES



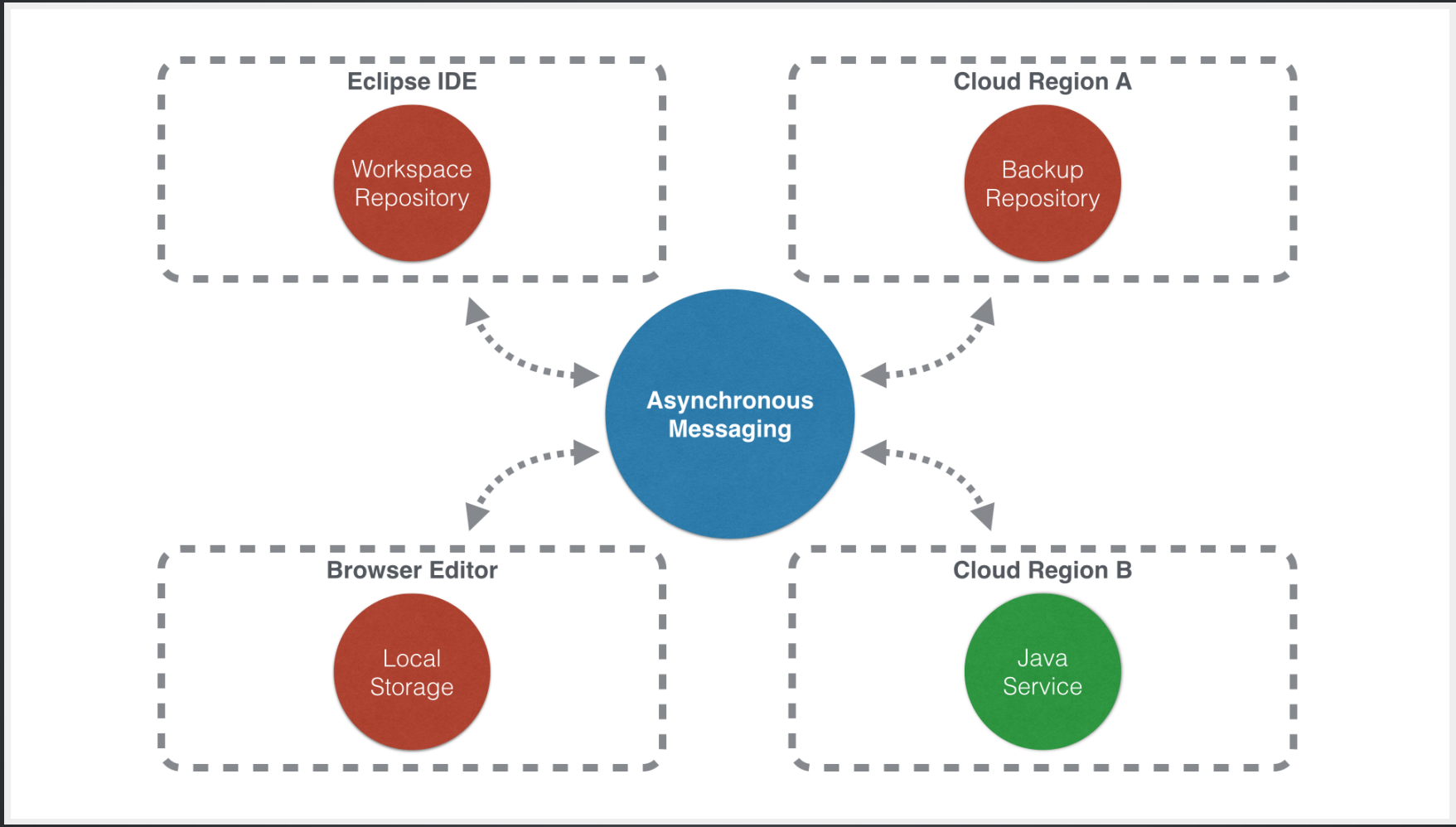
BROWSER & DESKTOP



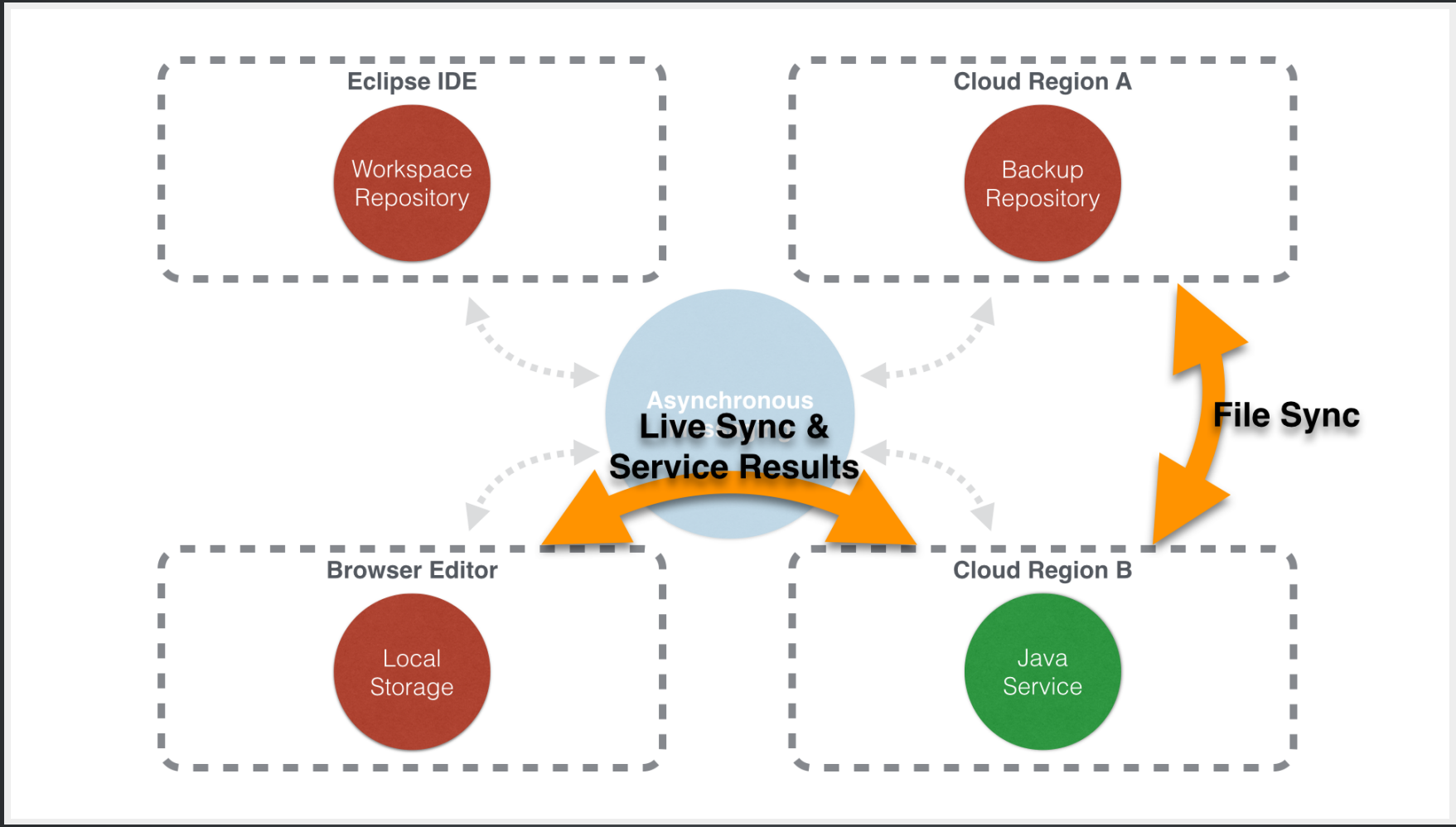
BROWSER & DESKTOP



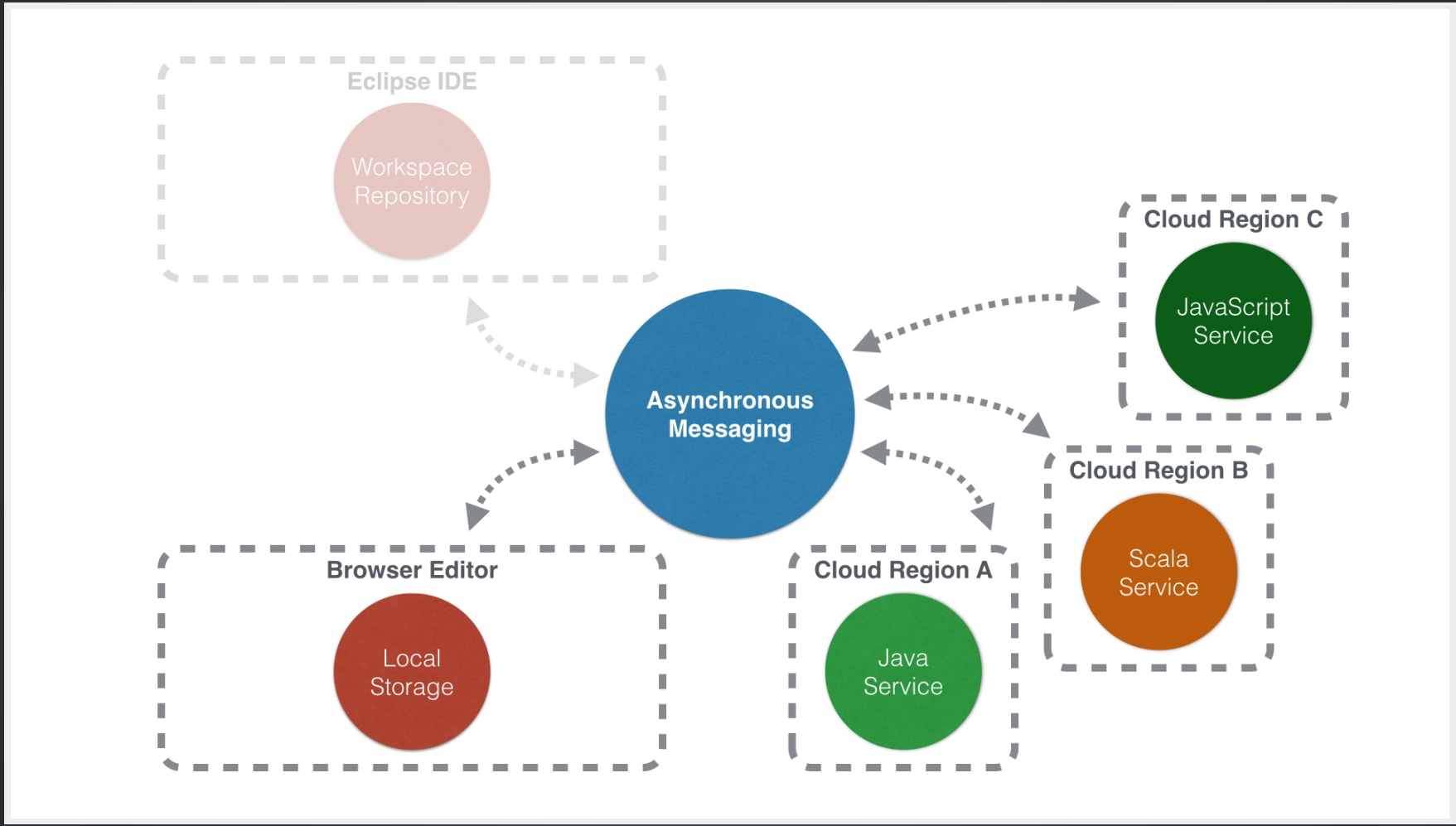
JAVA SERVICE IN THE CLOUD



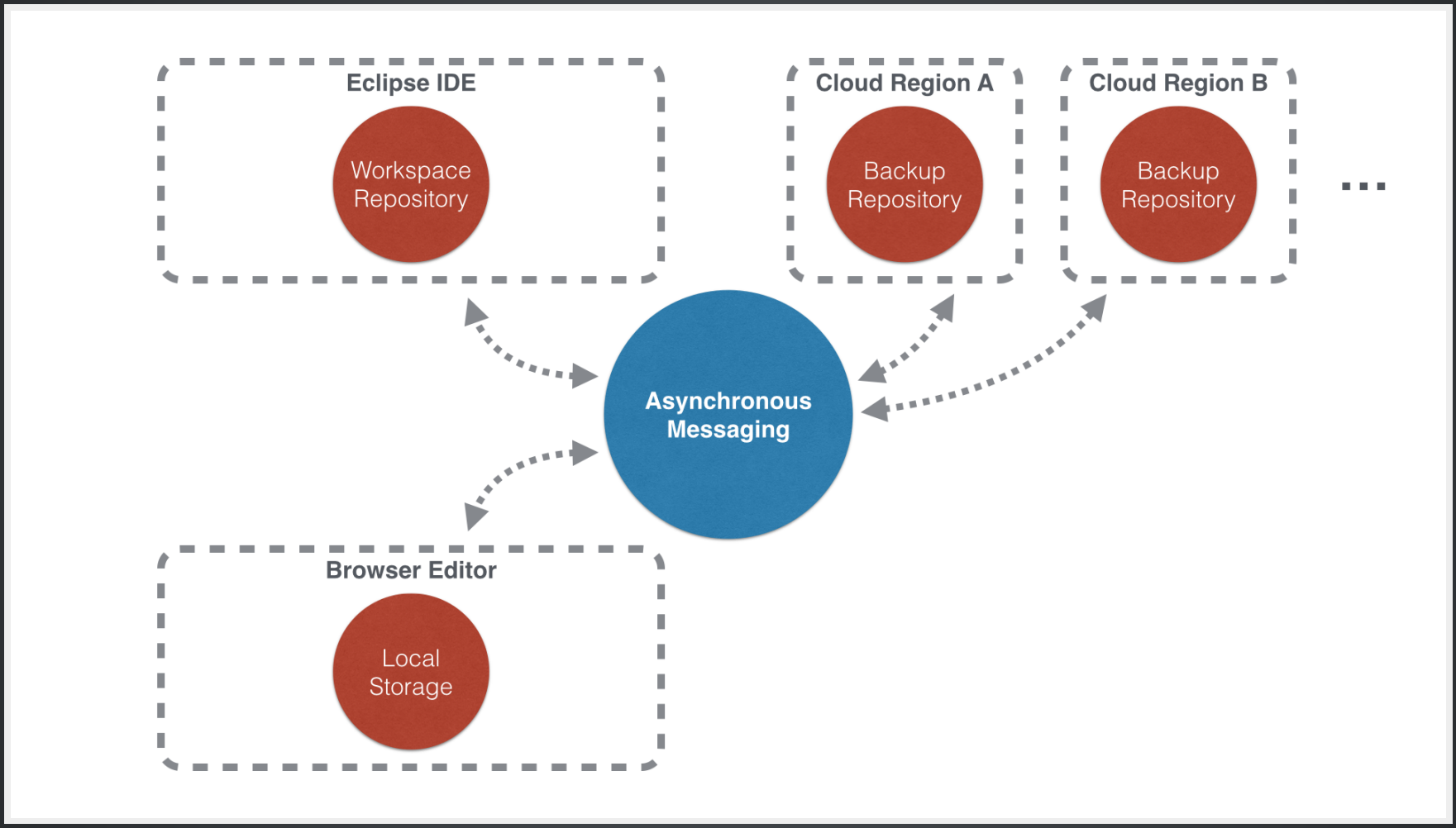
JAVA SERVICE IN THE CLOUD



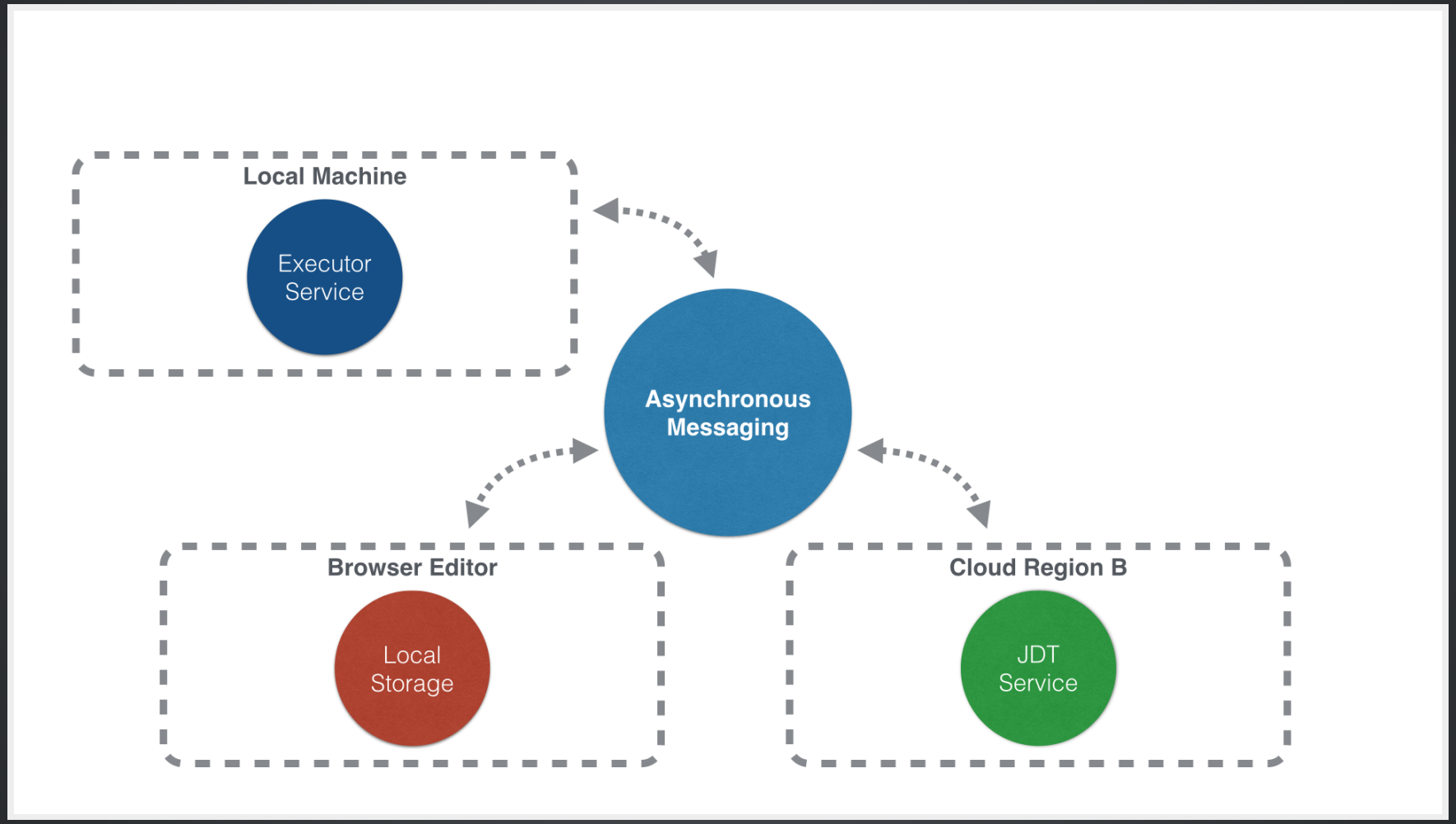
LANGUAGE SERVICES



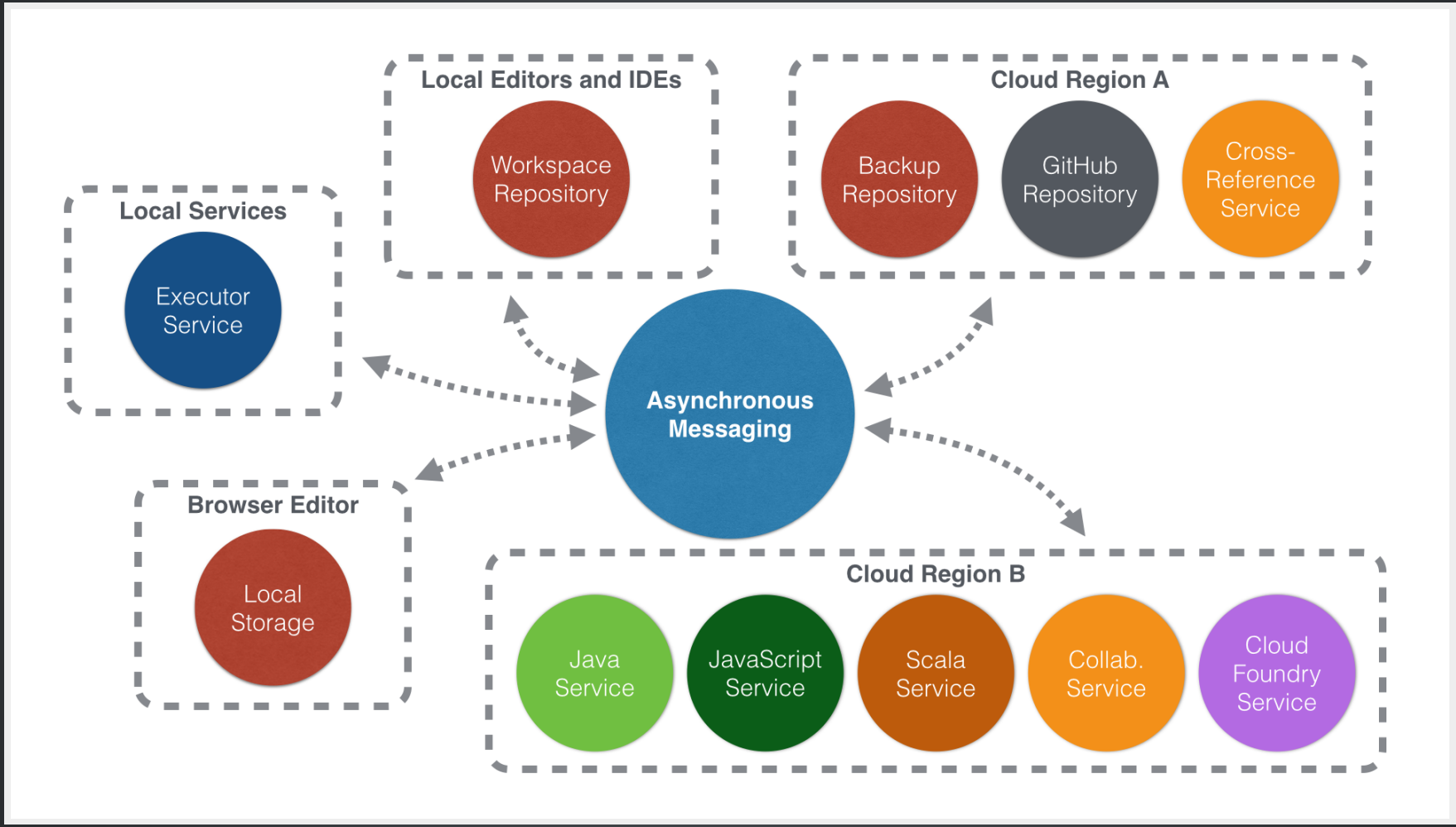
MULTIPLE BACKUP REPOSITORIES



LOCAL SERVICES EXAMPLE



THE VISION


Local Editors and IDEs
Workspace Repository
Cloud Region A
Backup Repository
GitHub Repository
Cross-Reference Service
Local Services
Executor Service
Asynchronous Messaging
Browser Editor
Local Storage
Cloud Region B
Java Service
JavaScript Service
Scala Service
Collab. Service
Cloud Foundry Service