Microsoft

Faster Inner Dev Loop For Stream Processing

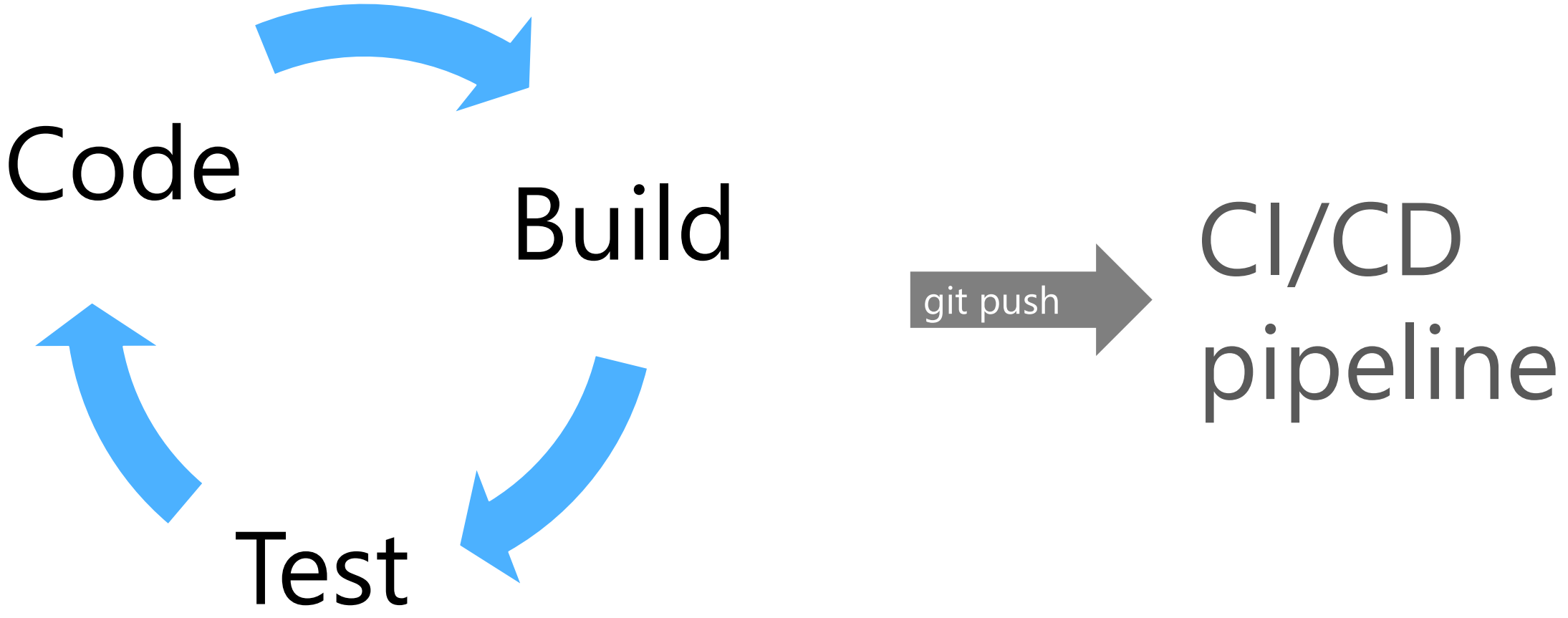
Florian Eiden, product manager, Azure Stream Analytics

2022-04

linkedin.com/in/fleid

@fleid_bi

The inner development loop should be fast



3 factors that make things slow

1. Distributed infrastructure (cloud or not)
2. Integration : strong coupling to sources and sinks
3. Extensive metadata definition

Workarounds

1. Distributed infrastructure **> local development**

   Pros : fastest experience, cheapest

   Cons : runtime desynch between local and cloud, not appropriate at scale, networking concerns

2. Integration : strong coupling to sources and sinks

3. Extensive metadata definition

Workarounds

1. Distributed infrastructure

2. Integration **> mocking sources with files**

   Pros : fastest, cheapest, no setup

   Cons : only an emulation, beware when switching back to live

3. Extensive metadata definition

Workarounds

1. Distributed infrastructure

2. Integration

3. Extensive metadata definition
    > **no DDL (schema) requirement**
    > **opinionated default settings**

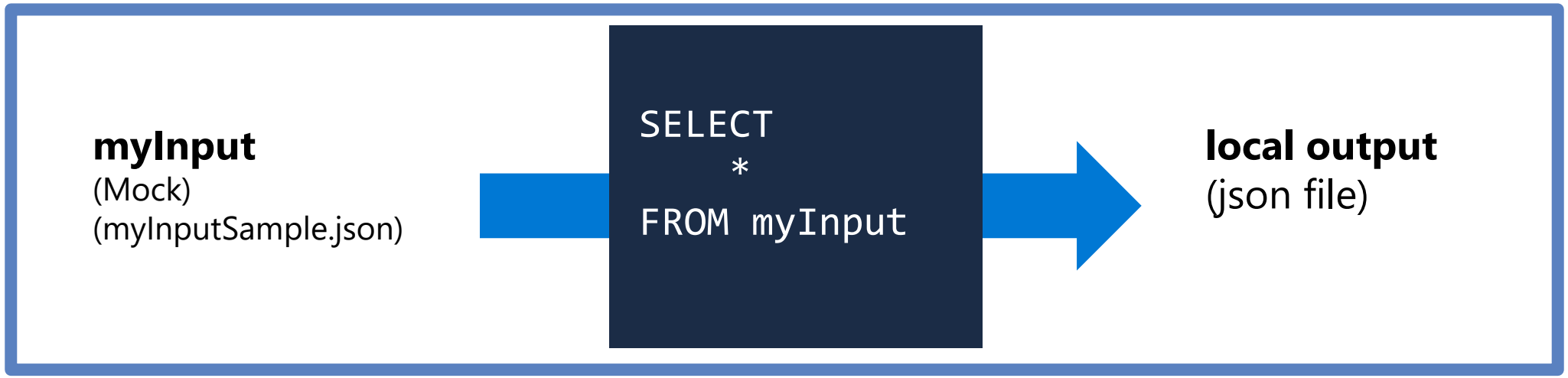
Pros : fastest time to query result, easy to explore the content of a stream

Cons : easy to rely on implicit conversions that will bite in production

Azure Stream Analytics – local experience

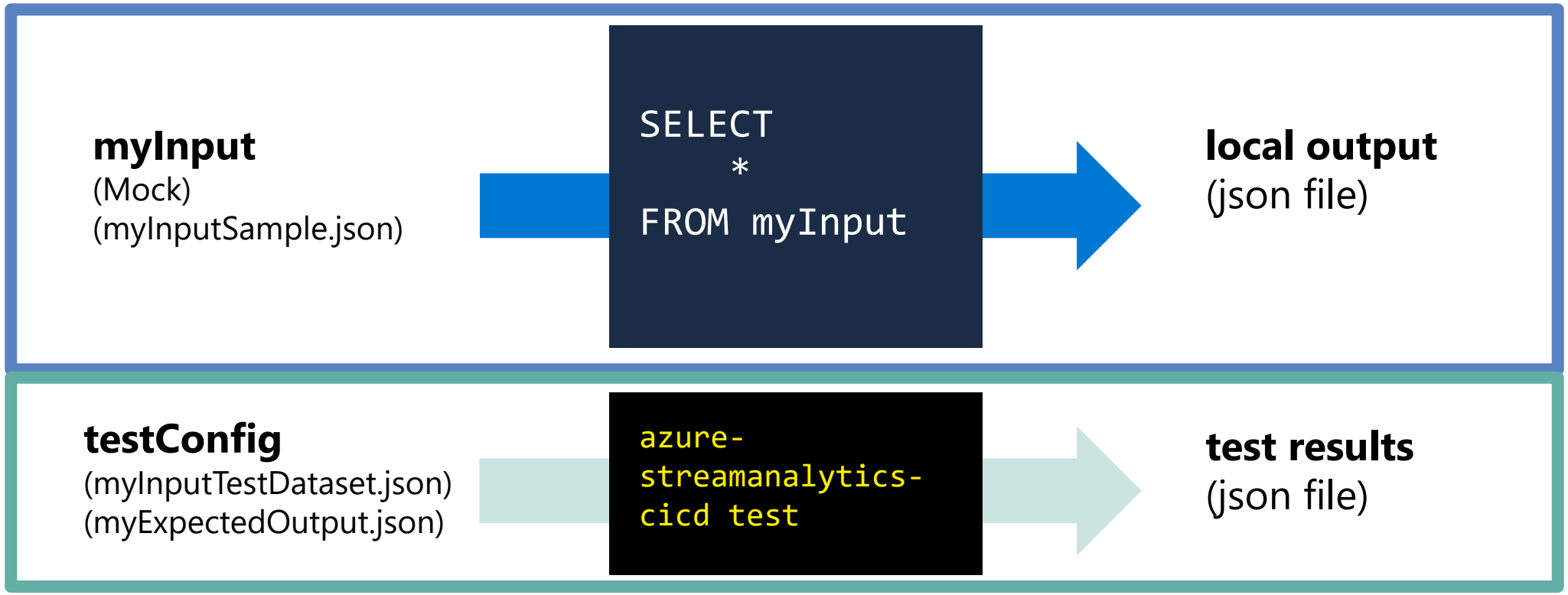
VS Code + Azure Stream Analytics Tools extension

> Local runs on mock input files



Azure Stream Analytics – unit testing

azure-streamanalytics-cicd npm companion package



Demo

Microsoft

Thank you

web	azure.com/sa
email	askasa@microsoft.com
Twitter	@AzureStreaming
GitHub	azure/azure-stream-analytics
Stack Overflow	azure-stream-analytics