Moving ABAP workloads to Serverless

Lars Hvam, Heliconia Labs, September 2021

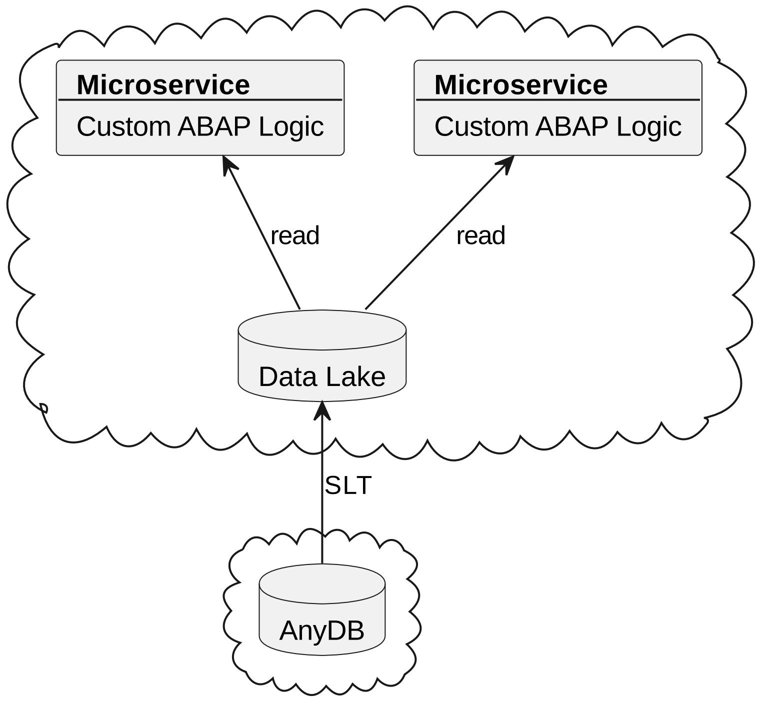
Home	https://github.com/heliconialabs/moving-abap-workloads-to-serverless
License	Creative Commons Attribution 4.0 International
Build	2022-10-16 11:15:05 UTC

1. Introduction

Sometimes microservices can help fulfilling business requirements, sometimes monoliths, both has its perils.

ABAP has traditionally only targeted monolithic applications, this paper suggests a multi-target setup where read-only ABAP code can be deployed to run on both traditional ABAP stacks and as a microservice on hyperscalers.

1.1. Setup



SLT replicates data to a data lake.

Data in the data lake is stored in raw format.

Custom ABAP logic runs as a microservice reading data from the data lake and can be accessed via standardized APIs. In order to run ABAP logic as a microservice, the ABAP code is transpiled to JavaScript.

1.2. Benefits

- API first
- Scaling up on cloud
- Scaling down on central OLTP system
- Deploy changes independently

1.3. Microservice Platforms

To avoid vendor lock-in, shims can be created, allowing to use the same ABAP logic on various platforms.

The ABAP code is transpiled to JavaScript, which is supported by most serverless platforms,

- Kyma Serverless
- OpenFaaS
- Kubeless
- OpenWhisk
- Microsoft Azure Functions
- Google Cloud Functions
- AWS Lambda
- Cloudflare Workers

2. Development

The ABAP code is maintained in a git repository, making heavy use of Continuous Integration to ensure compatibility and correct functionality.

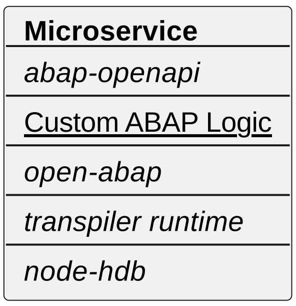
The transpiler accepts ABAP 7.02 language syntax, but the ABAP logic can be maintained in higher language syntax and then automatically downported as part of the build and test process.

The continuous integration platform can build and test every change(commit), perform unit tests, perform integration tests, compile, build OCI images if needed.

Developers can build the code locally using vscode or similar, plus run unit and integration tests on local copies before commit.

2.1. Microservice Stack

The microservice itself contains the ABAP business logic, along with multiple other components allowing the code to run and connect/provide services to other systems.



The stack consists of both transpiled and native JavaScript code

- Transpiled

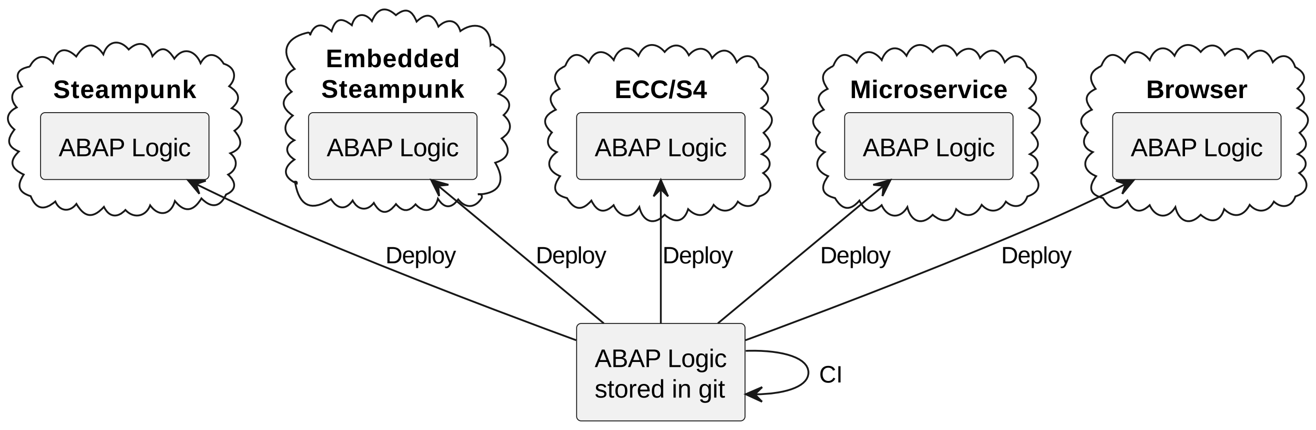
  - abap-openapi, allows easily exposing OpenAPI REST services from ABAP
  - Custom ABAP Logic
  - open-abap, reuseable ABAP functionality, connecting ABAP and JS features
- Native JS
  - @abaplint/runtime
  - node-hdb

2.2. Multi target ABAP applications

If dependencies are kept under control, the ABAP code can be deployed to any platform.

Sometimes microservice are beneficial, but if Embedded Steampunk is introduced in the landscape, it can be considered if the code should be moved to that platform instead.

Building multi target ABAP applications allows deploying to any runtime, depending on business requirements.


Steampunk
ABAP Logic
Embedded Steampunk
ABAP Logic
ECC/S4
ABAP Logic
Microservice
ABAP Logic
Browser
ABAP Logic
Deploy
Deploy
Deploy
Deploy
Deploy
ABAP Logic stored in git
CI