

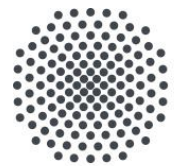
# Combining the Best of Two Worlds: Microservices and Micro Frontends as Basis for a new Plugin Architecture



**Fabian Bühler, Johanna Barzen, Lukas Harzenetter,  
Frank Leymann, Philipp Wundrack**

*{buehler, barzen, harzenetter, leymann, wundrack}*  
*@iaas.uni-stuttgart.de*

Institute of Architecture of Application Systems



University of Stuttgart



PlanQK

Supported by:



Federal Ministry  
for Economic Affairs  
and Climate Action

on the basis of a decision  
by the German Bundestag

# Use-Case QHAna

One-Hot-Encoding

Wu-Palmer Similarity

SVM

kNN



Autoencoder

QaoaMaxCut

## QHAna

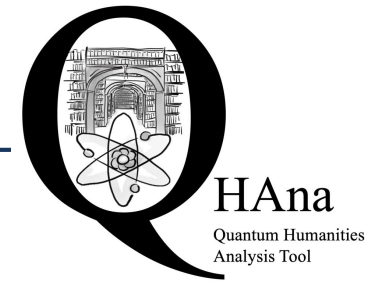
Quantum Humanities  
Analysis Tool

qiskitQSVC

KMeans

Multidimensional Scaling

# Use-Case QHAna



## Algorithms:

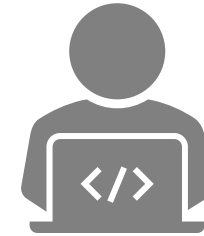
One-Hot-Encoding  
Wu-Palmer Similarity  
Multidimensional  
Scaling  
QaoaMaxCut  
kNN  
KMeans  
SVM  
qiskitQSVC  
Autoencoder

## SDKs:

- Qiskit
- Forest
- Cirq
- Strawberry Fields
- QUIL
- Silq
- Q#
- ...

## Users:

- Digital Humanities
- GUI

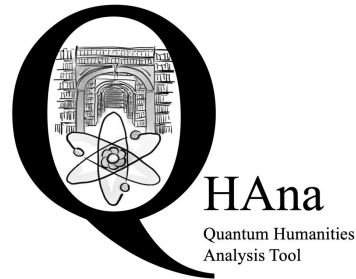


## Reusability:

- Workflows
- Standalone
- Research

# Use-Case QHAna

---



- Many algorithms with more to come  $\Rightarrow$  Extensible plugin architecture
- Usability for digital humanities audience  $\Rightarrow$  GUIs for algorithms
- Reusability outside QHAna  $\Rightarrow$  (REST) APIs
- Different quantum SDKs & programming languages  $\Rightarrow$  Support all languages/SDKs

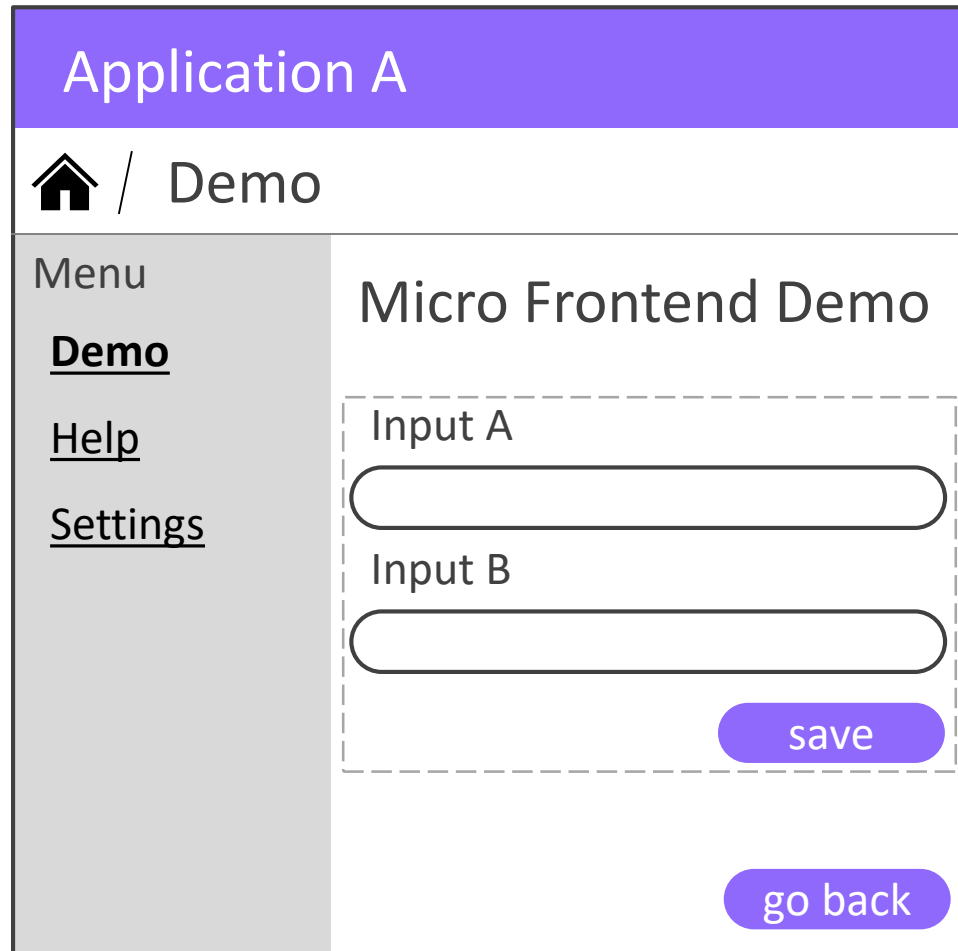
# Motivation

---

## Requirements:

- Extensible plugin architecture
- Plugins with GUIs
- Reusable plugins
- Support different programming languages

# GUI and Reusability



Application A

Home / Demo

Menu

- Demo**
- Help
- Settings

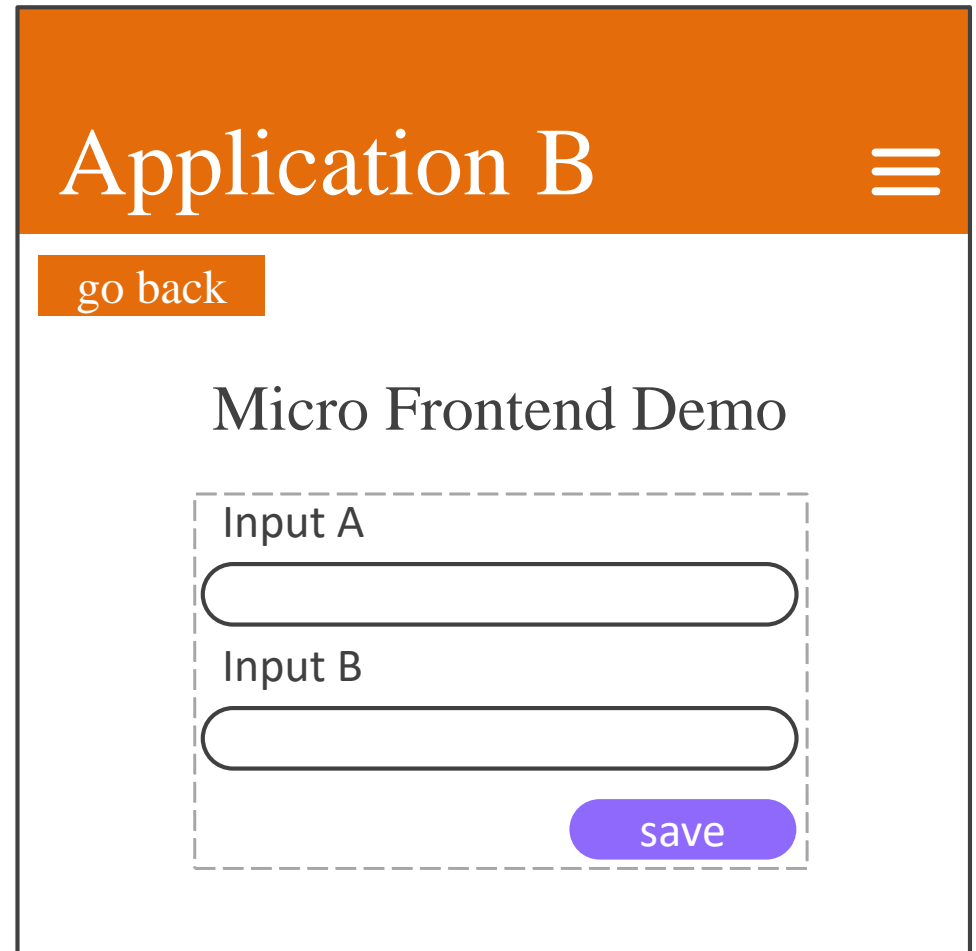
Micro Frontend Demo

Input A

Input B

save

go back



Application B

go back

Micro Frontend Demo

Input A

Input B

save

# Motivation

---

## Requirements:

- Extensible plugin architecture
- Plugins with GUIs
- Reusable plugins **& plugin GUIs**
- Support different programming languages

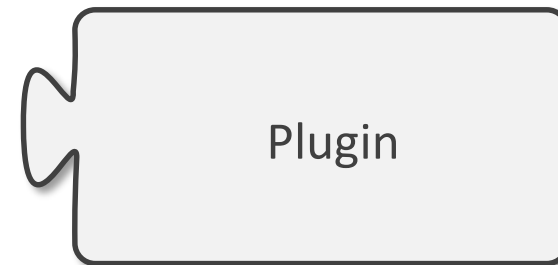
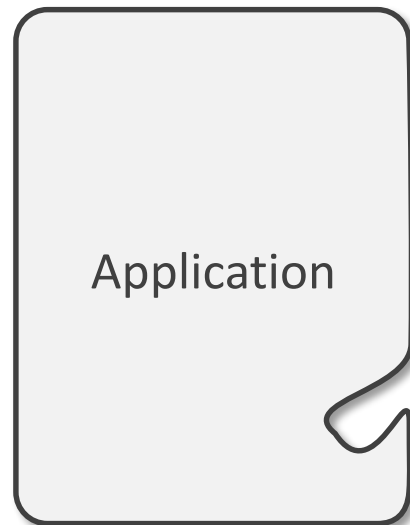
How?



# Background – Plugins

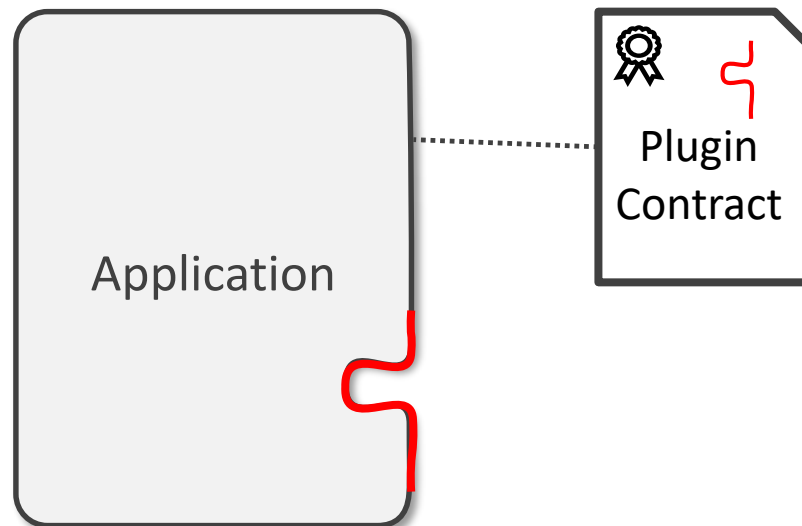
---

- Can extend applications with new functionality
- Developed independently

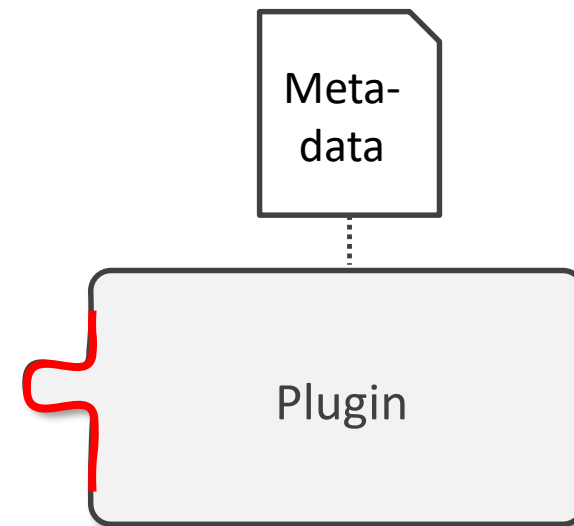


# Background – Plugins

- Can extend applications with new functionality
- Developed independently
- Automated integration by an end-user
- Powerful extension mechanism
- Developed for specific Application



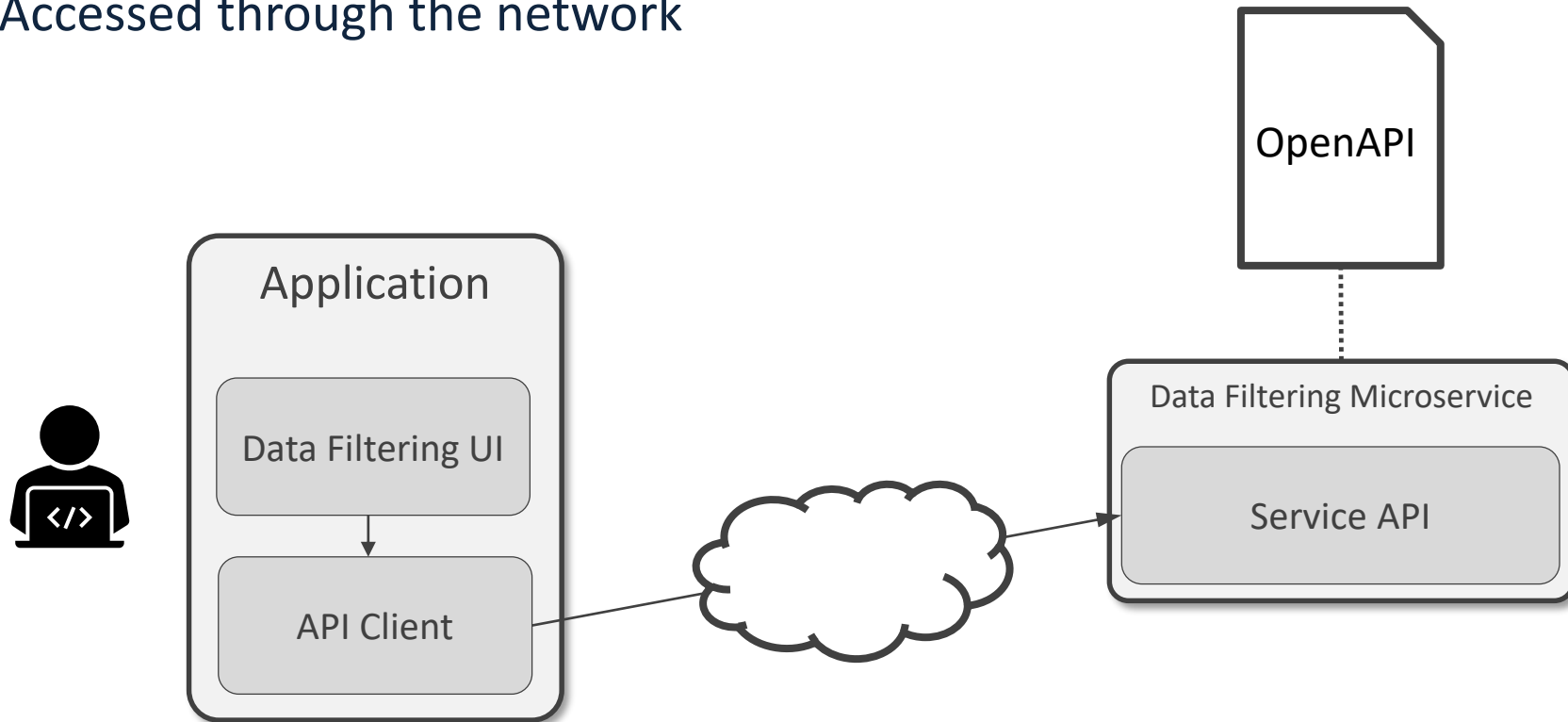
Extensible Architecture ✓



# Background – Microservices

- Independent application
- Exposes an API
- Integrated by a developer
- Accessed through the network

Reusable Components ✓



# Plugins vs. Microservices

---

## Plugins

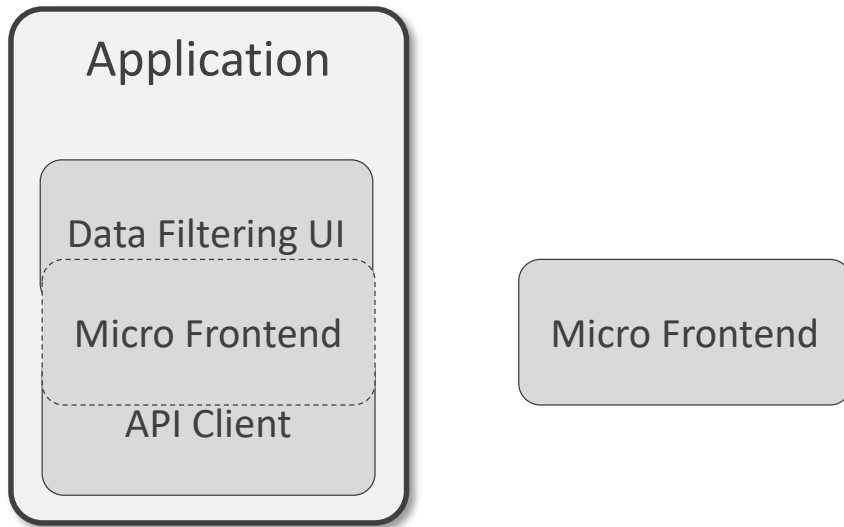
- **Powerful extension mechanism** ★
- **Automated integration** ★
- Developed for specific Application
- Runs inside application

## Microservices

- Service API
- Manual integration
- **Reusable** ★
- **Independent** ★

Support any programming language ✓

# Background – Micro Frontends

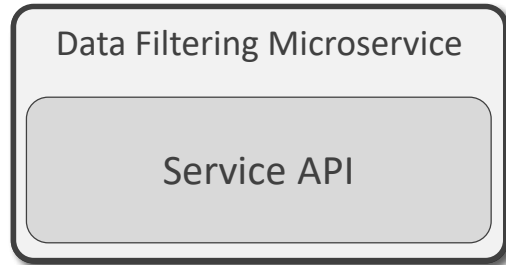


GUIs ✓

- A way to break up GUIs
- Based on HTML
- Independent development & deployment
- Server-side integration
- Client-side integration
  - iframe
  - JavaScript
    - Webcomponent

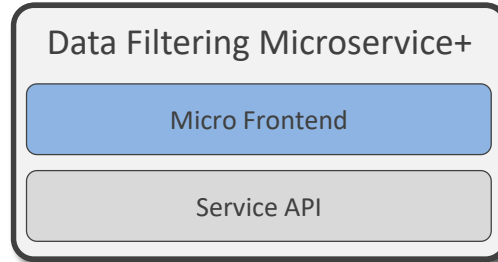
Lets combine the building blocks

# Reusable Microservice Based Plugins



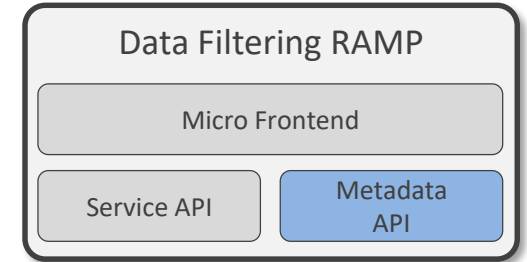
Microservice

- **Reusable API**
- No GUI



Microservice with  
Micro Frontend

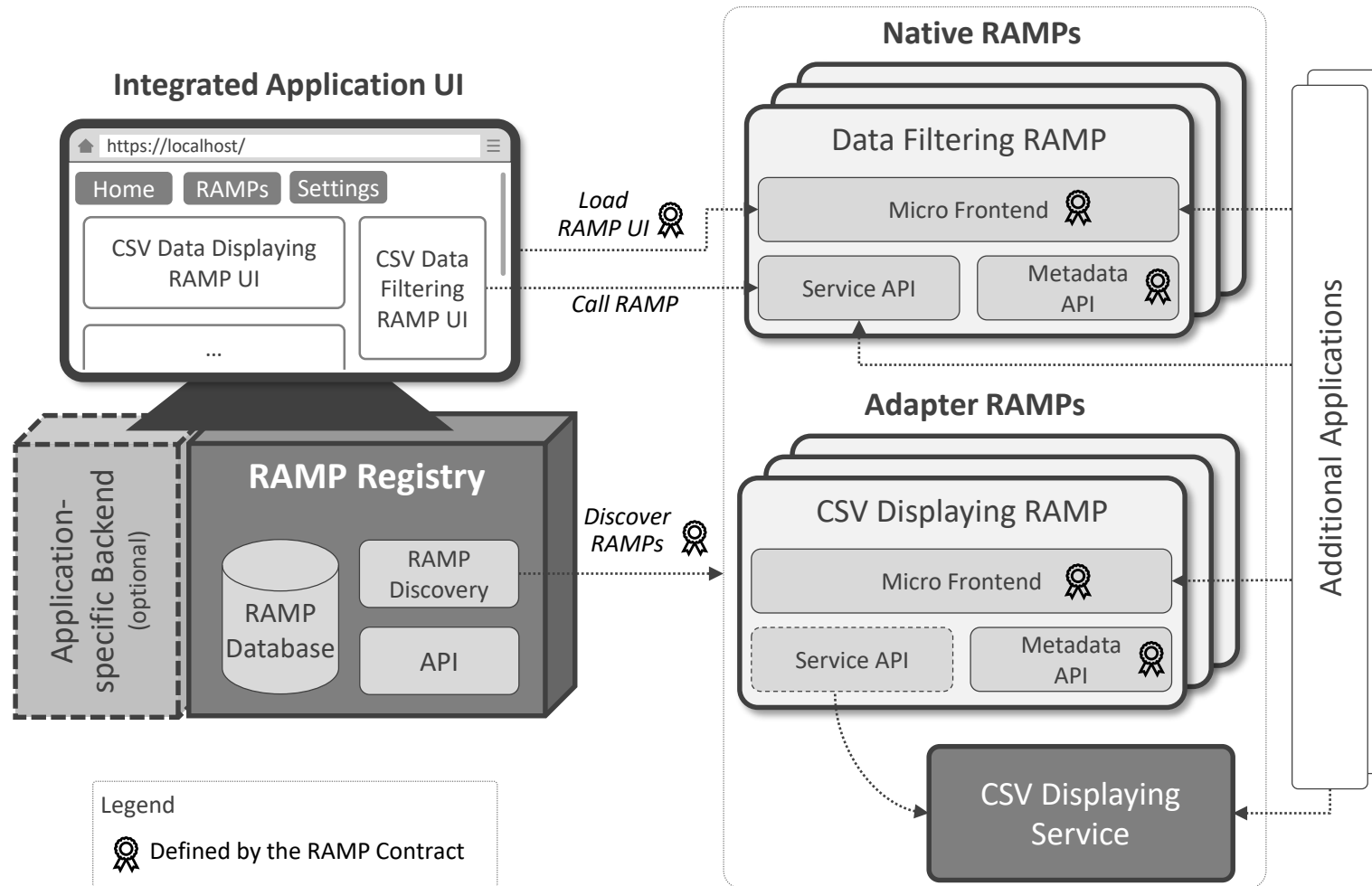
- Reusable API
- **Micro Frontend**



Reusable Microservice  
Based Plugin (RAMP)

- Reusable API
- Micro Frontend
- **Metadata**

# RAMP Architecture





# Micro Frontend Styling

## Application A

🏠 / Demo

Menu

- Demo**
- [Help](#)
- [Settings](#)

### Micro Frontend Demo

Input A

Input B

save

go back

## Application B

☰

go back

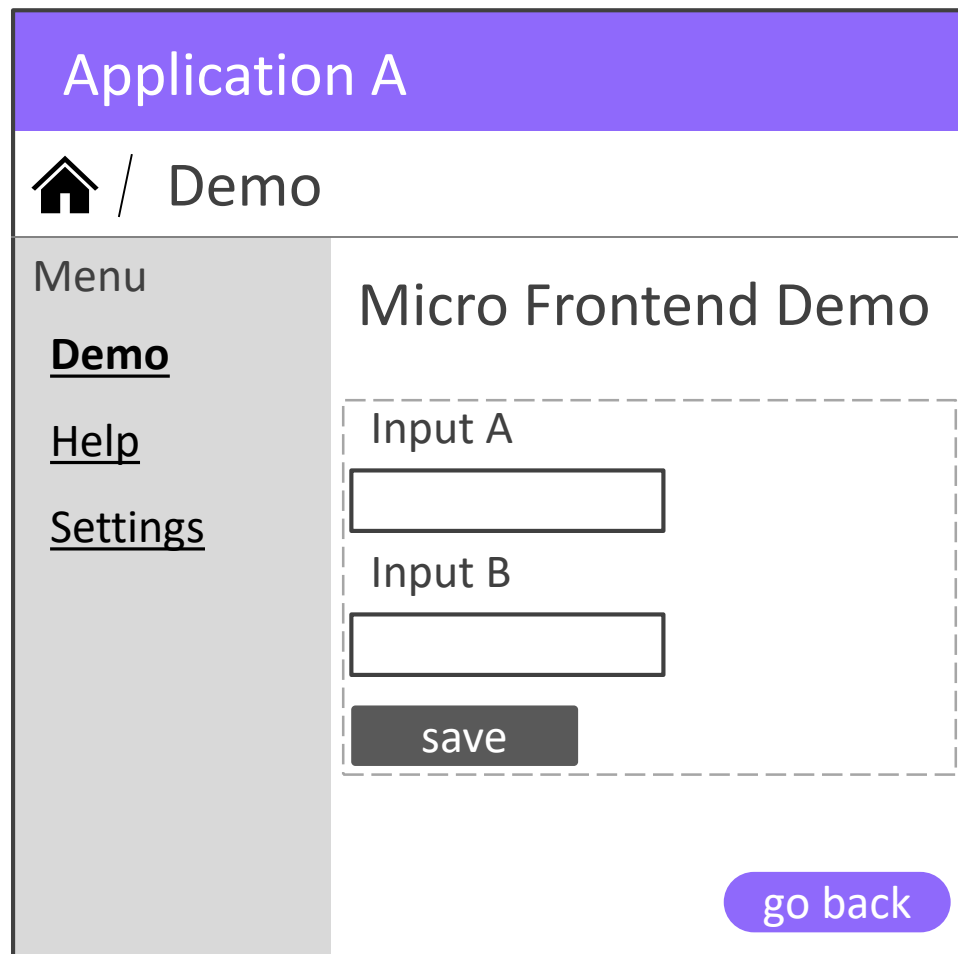
### Micro Frontend Demo

Input A

Input B

save

# Micro Frontend Styling



Application A

Home / Demo

Menu

- Demo**
- Help
- Settings

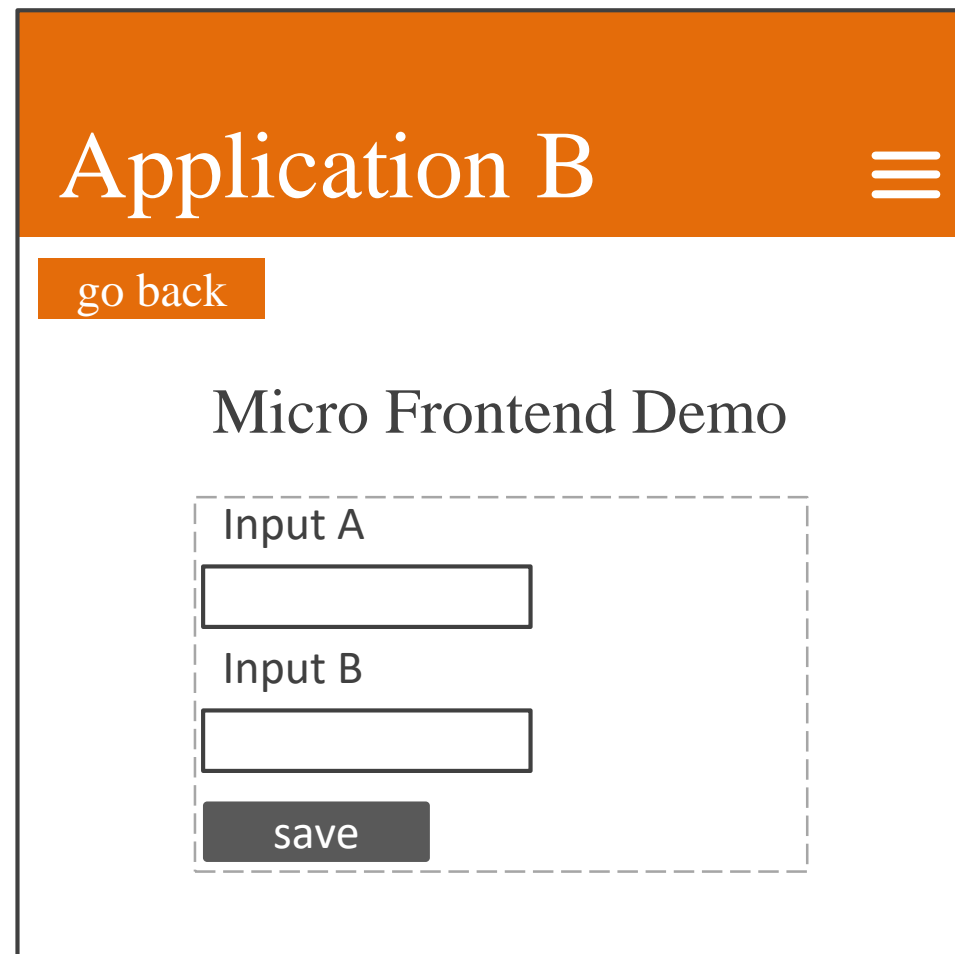
Micro Frontend Demo

Input A

Input B

save

go back



Application B

go back

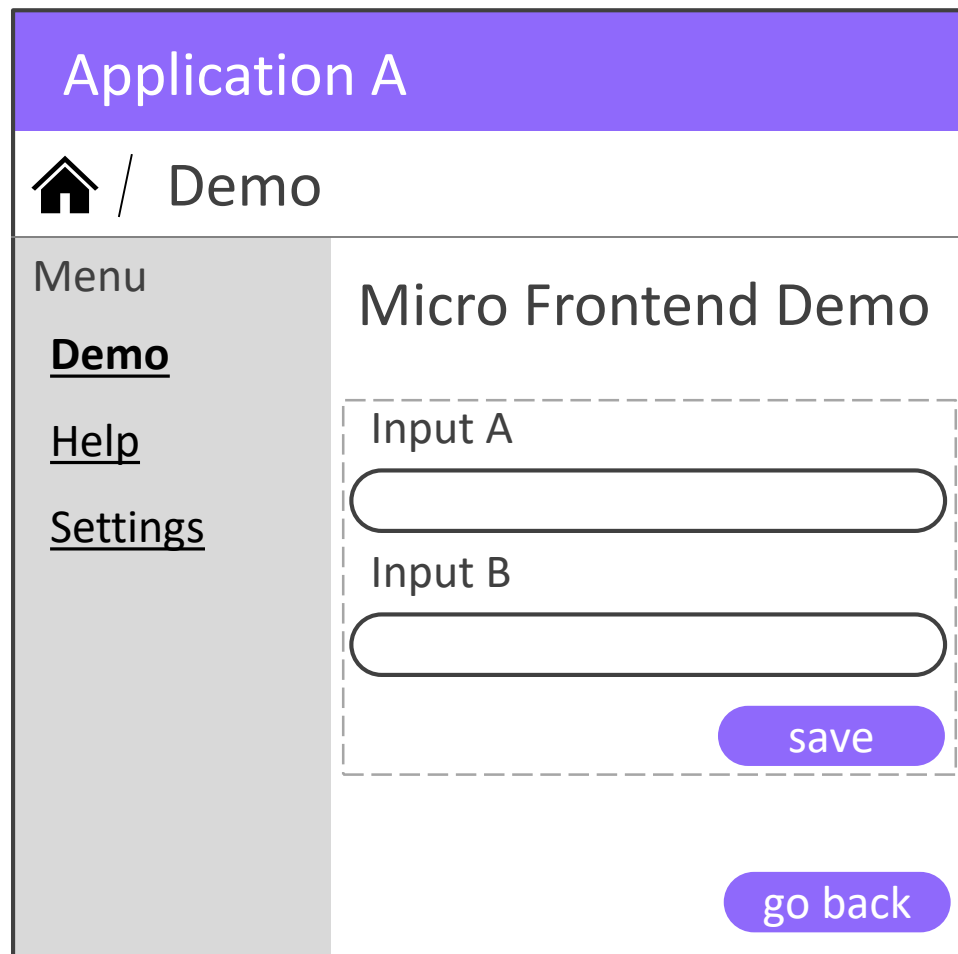
Micro Frontend Demo

Input A

Input B

save

# Micro Frontend Styling



Application A

Home / Demo

Menu

- Demo**
- Help
- Settings

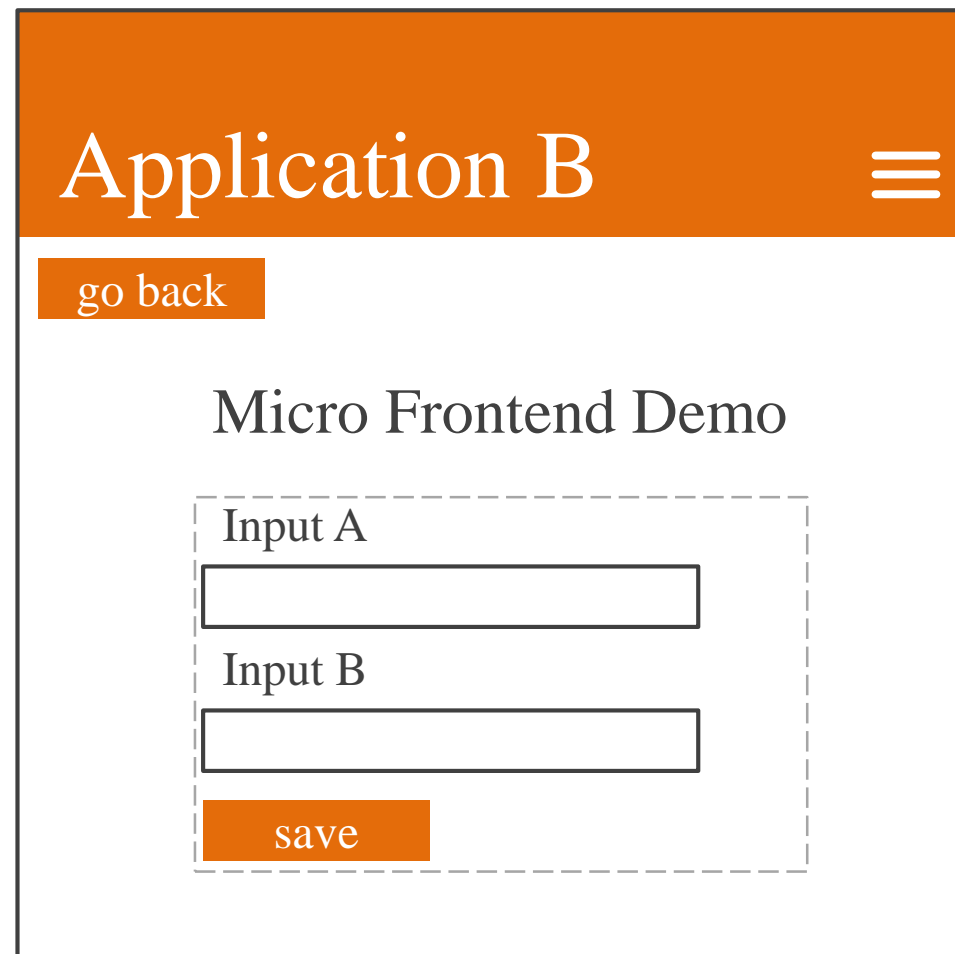
Micro Frontend Demo

Input A

Input B

save

go back



Application B

go back

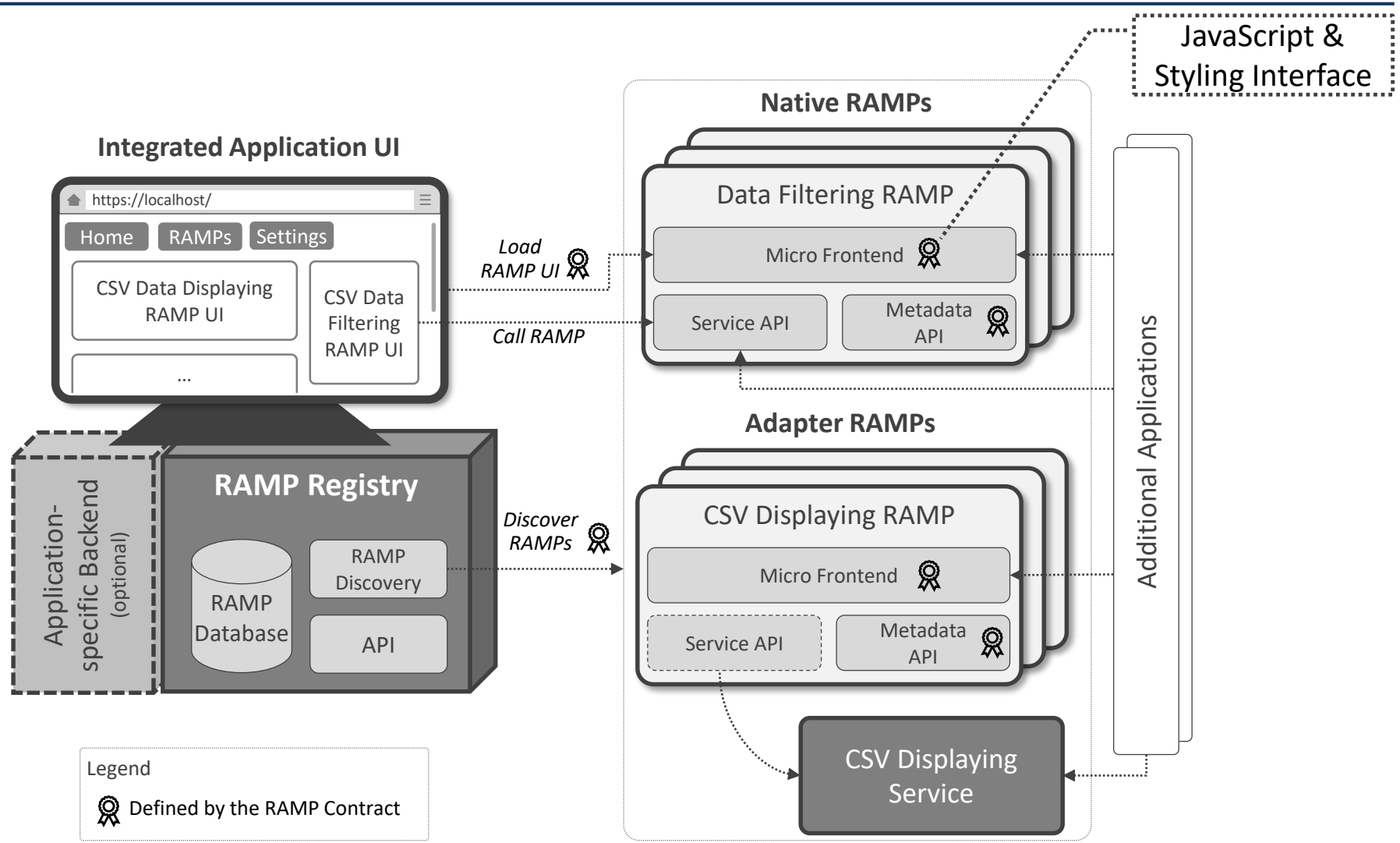
Micro Frontend Demo

Input A

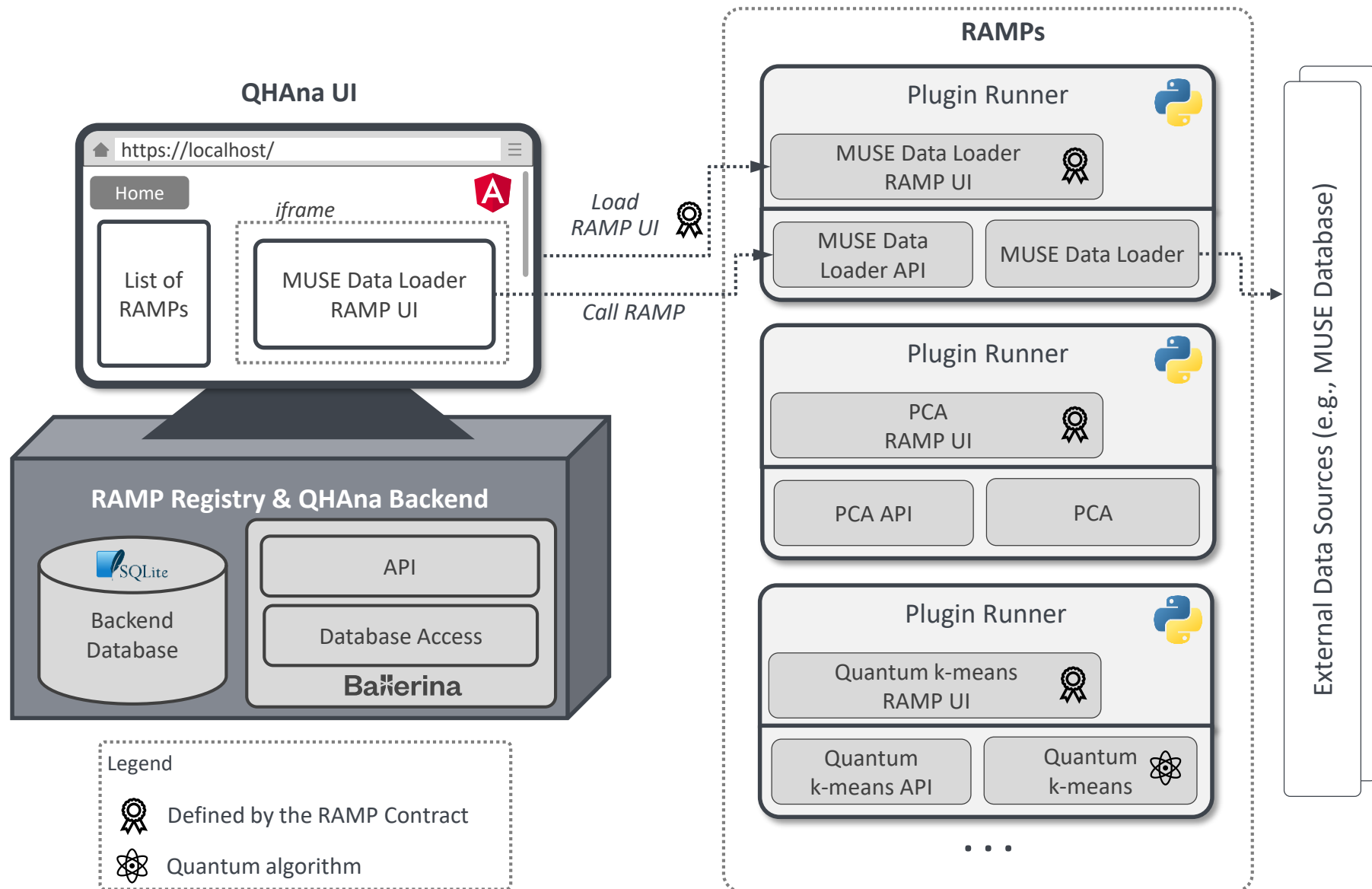
Input B

save

# RAMP Architecture



# Use-Case QHAna



# When to Use RAMPs

---

- Need plugins, but reusable
  - e.g. with multiple applications
- Have microservices, but need plugins
- Plugins for web applications
- Multiple programming languages

# Conclusion and Future Work

---

- RAMPs = Microservice + Micro Frontend + Metadata
  - RAMPs as microservices  $\Rightarrow$  reusability
  - RAMPs as plugins  $\Rightarrow$  automated integration
    - UI integration through styling interface
- RAMPs architecture
- Concrete use-case: QHAna
- Future Work:
  - RAMPs interacting with other RAMPs
  - RAMPs composed of RAMPs

Thank you!