Introducing OAuth2 in Olog/Phoebus

EPICS Collaboration Meeting, 2025

Speaker: Giovanni Lorenzo Napoleoni@INFN-LNF

Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Frascati

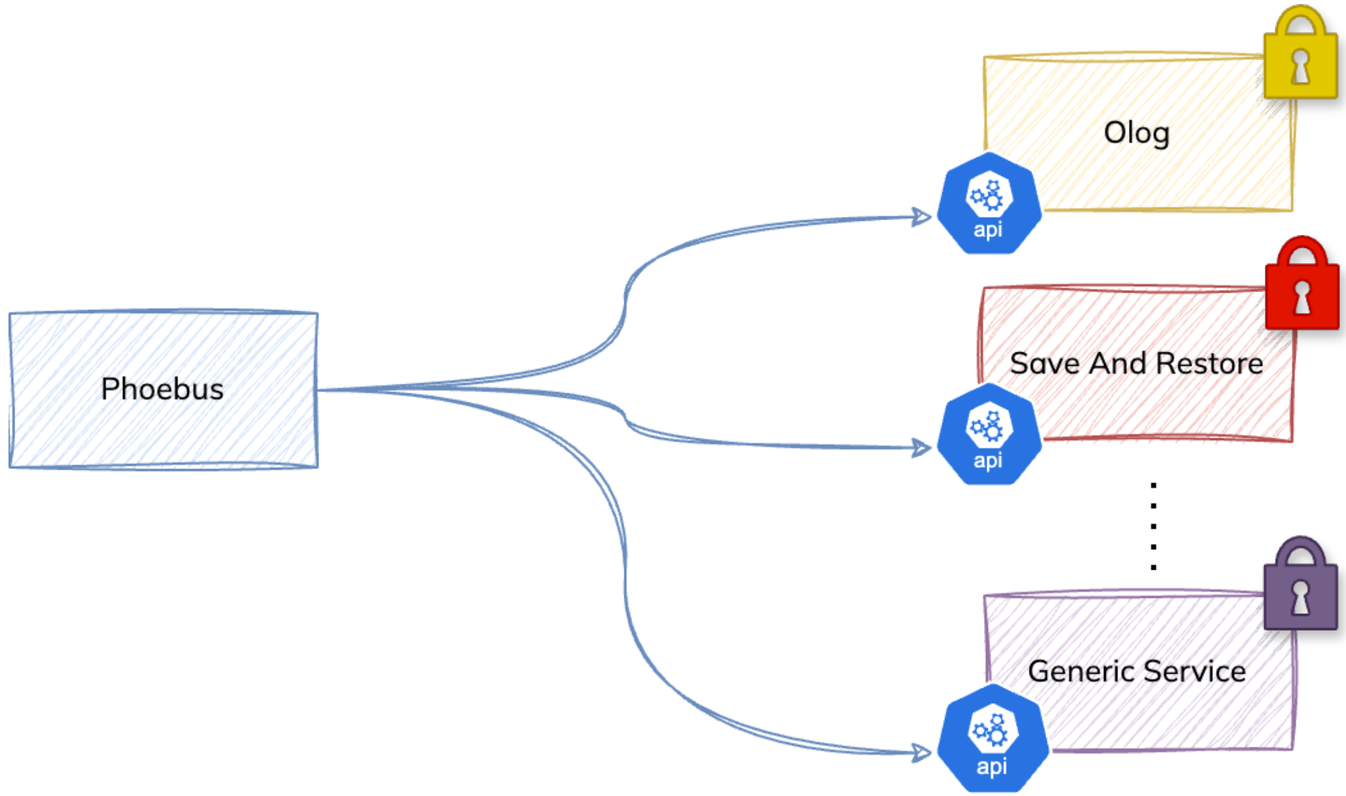
Phoebus Services Architecture

Different Services

Each service has its own API and authentication.

Authentication methods

Different services use different authentication and authorization protocols.



Problems

01

Handling flow

Need to implement the authentication/authorization flow separately for each service.

02

Credential Exposure

Each service must handle credential input.

03

Update authentication code

Maintaining different authentication mechanism is hard across multiple application

Solution

Use an auth protocol based on access delegation

One authentication/authorization flow for all services

+

Authorization **Authentication**

What is OAuth2

01

Authorization Framework

Clients are authorized without sharing password data with services

02

Token-Based Access Control

Clients use a **time-limited access** token to request protected resources.

03

API Security

Provides API security through scoped access tokens

What is OpenID

01

Decentralized Authentication Protocol

allows users to authenticate with multiple websites and applications using a single identity provider (IdP)

02

Based on OAuth 2.0

Enabling authentication in addition to authorization. It provides user identity information in a secure and standardized way.

03

Single Sign-On

Allowing users to log in seamlessly across different platforms with providers like Google, Microsoft, and Keycloak.

Why need both?

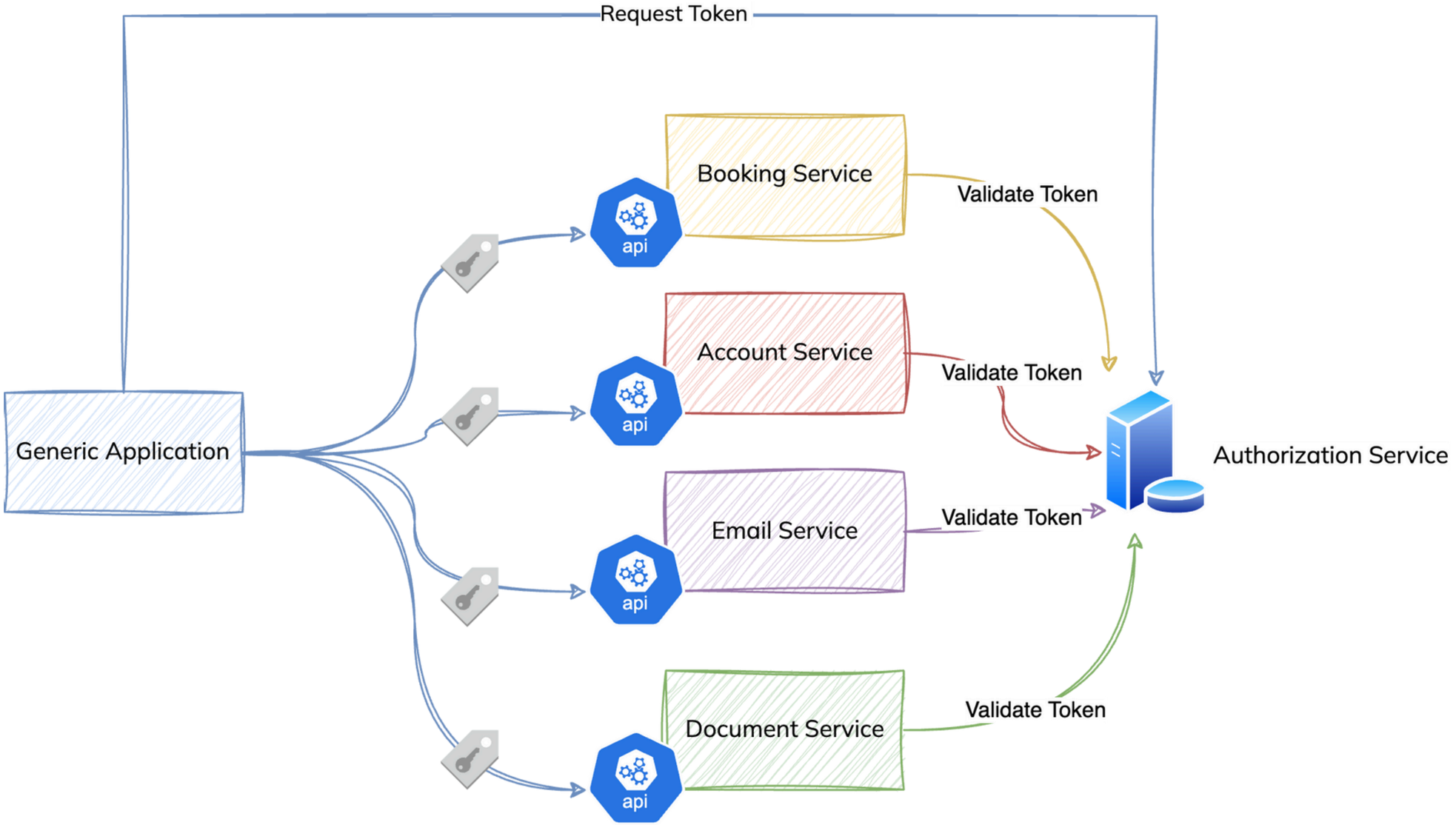
OAuth 2.0

Can the user access a resource?

OpenID Connect

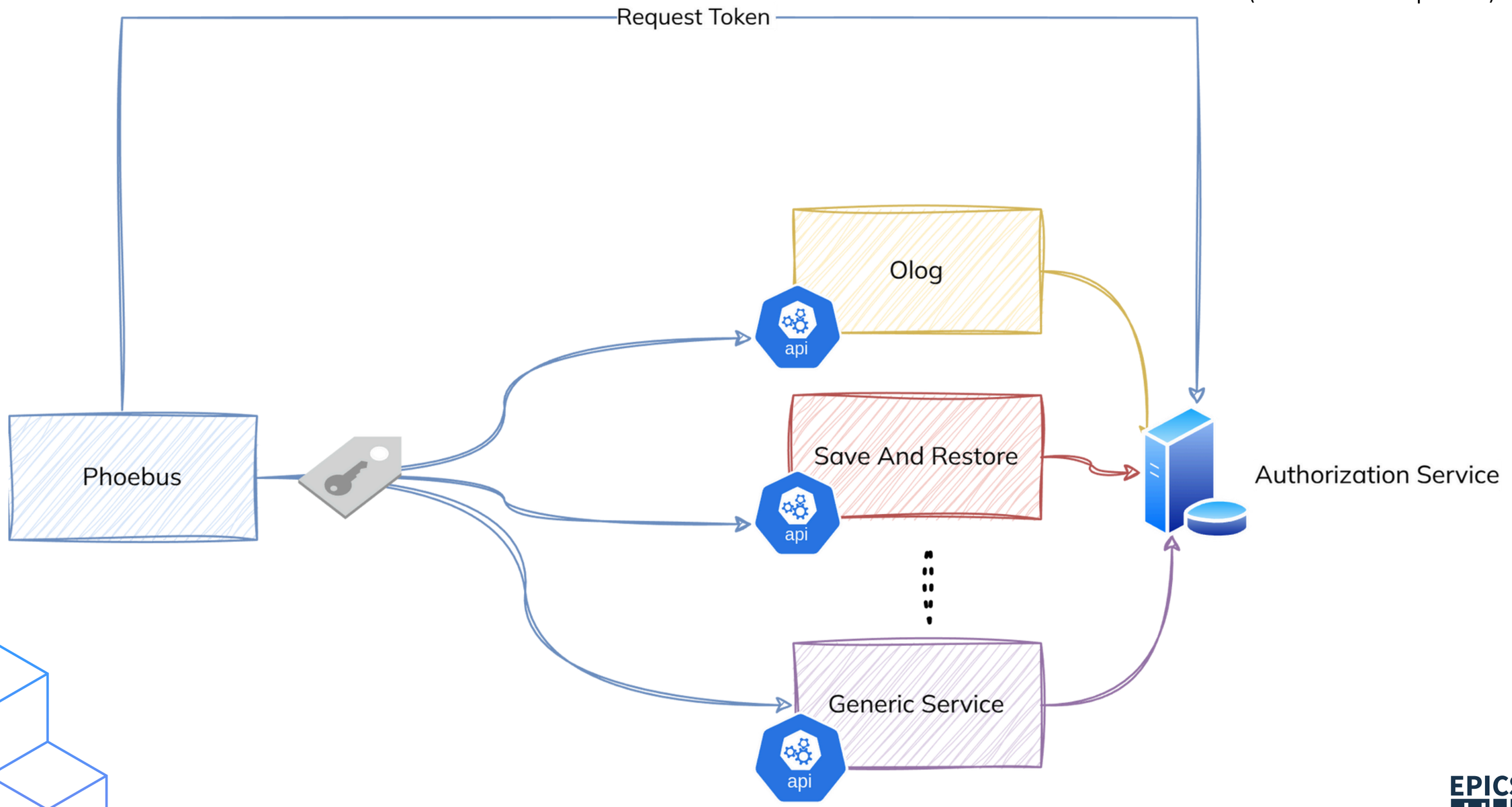
Who is the user accessing the resource?

INFN service architecture

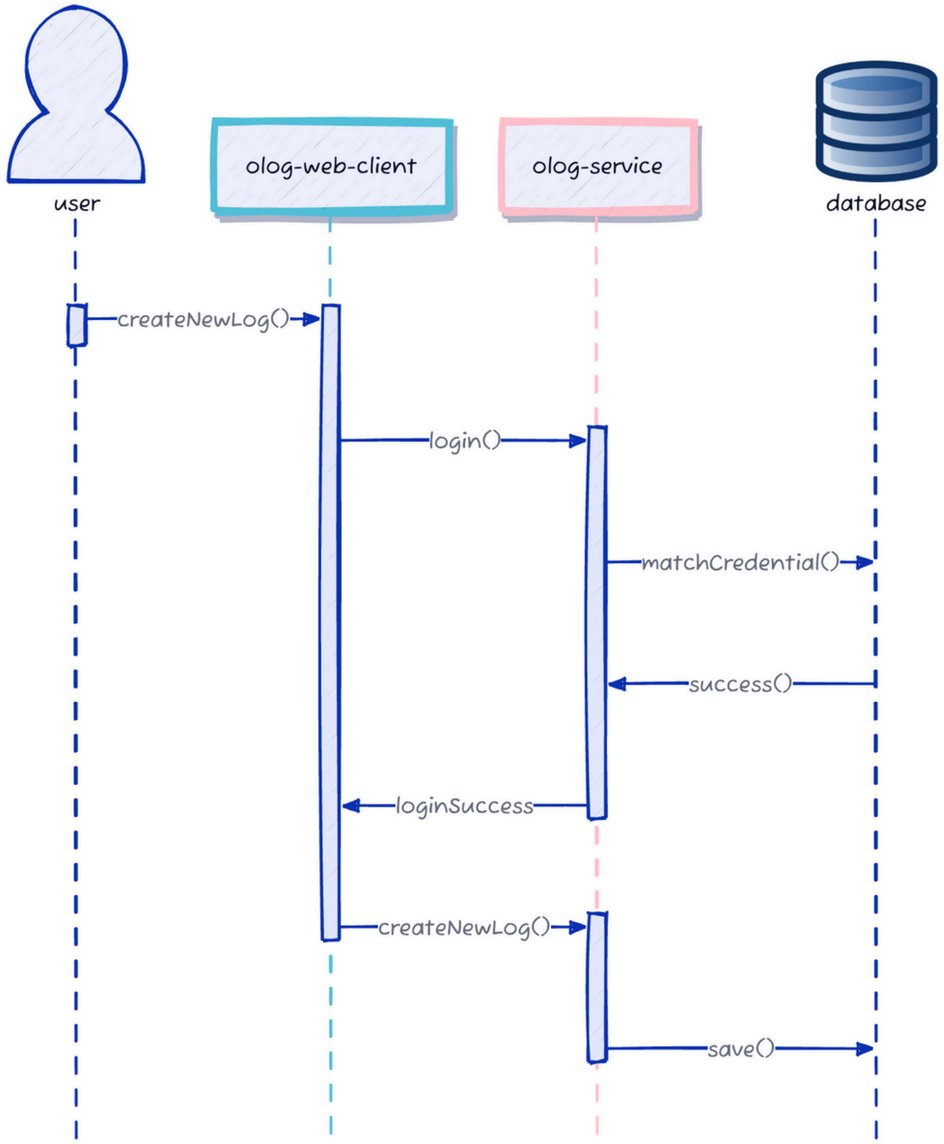


Phoebus service architecture

(Current development)



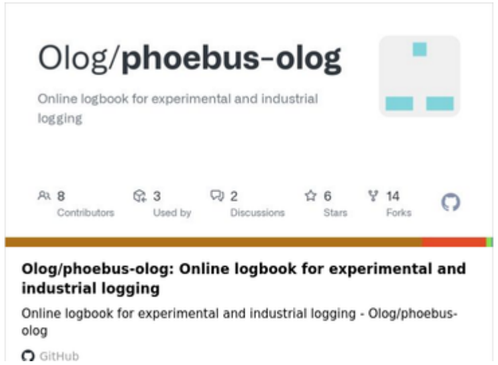
Use case with Olog



Create new log

User login with credentials stored on olog-service DB

Olog web client implements login forms for authentication

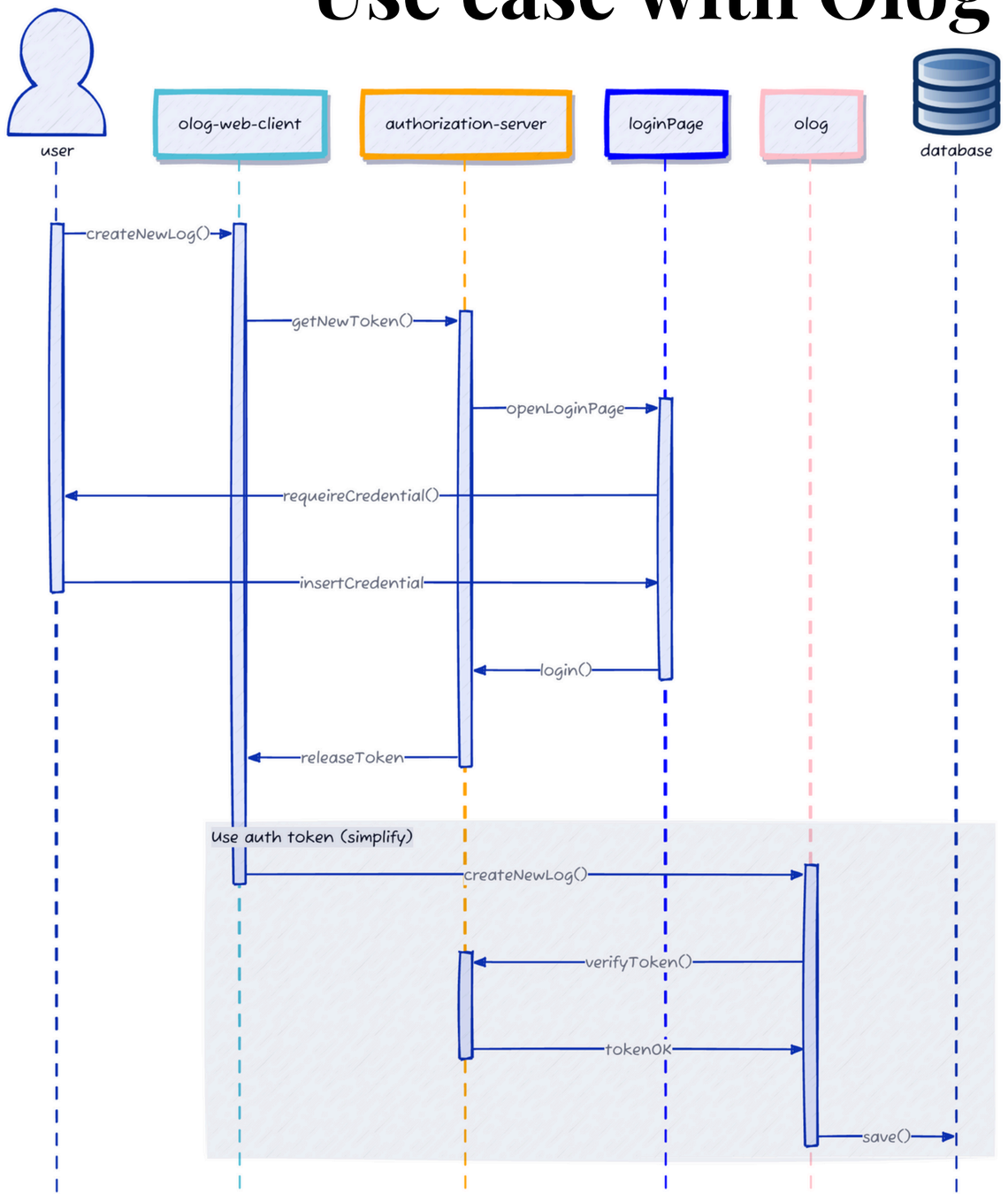


Use case with Olog (2)

Create new log

OAuth2 Implementation

Users authenticate via an Authorization Server and use token for API requests.

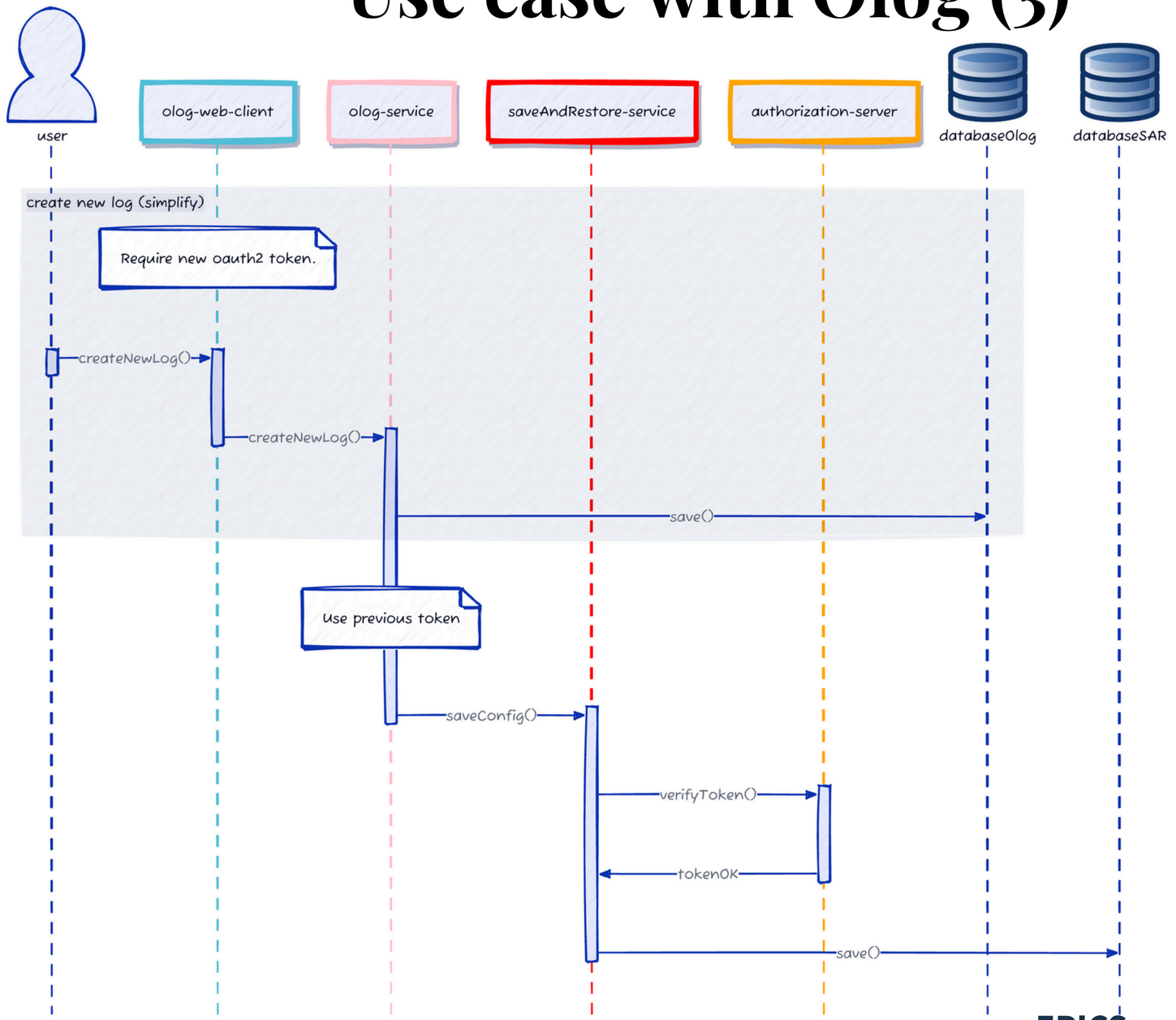


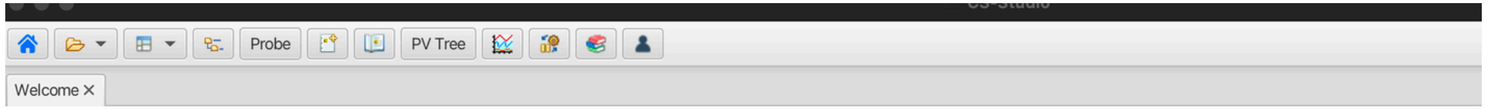
Use case with Olog (3)

New log with SaR

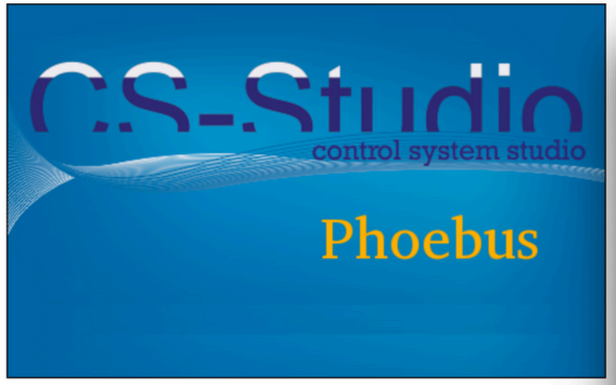
OAuth2 Implementation

After create a new log, save log configuration to save-and-restore service without requiring re-authentication.



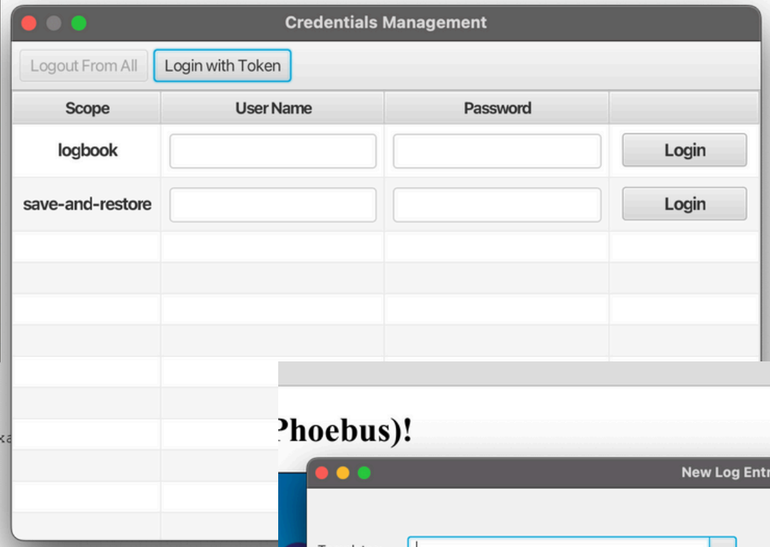


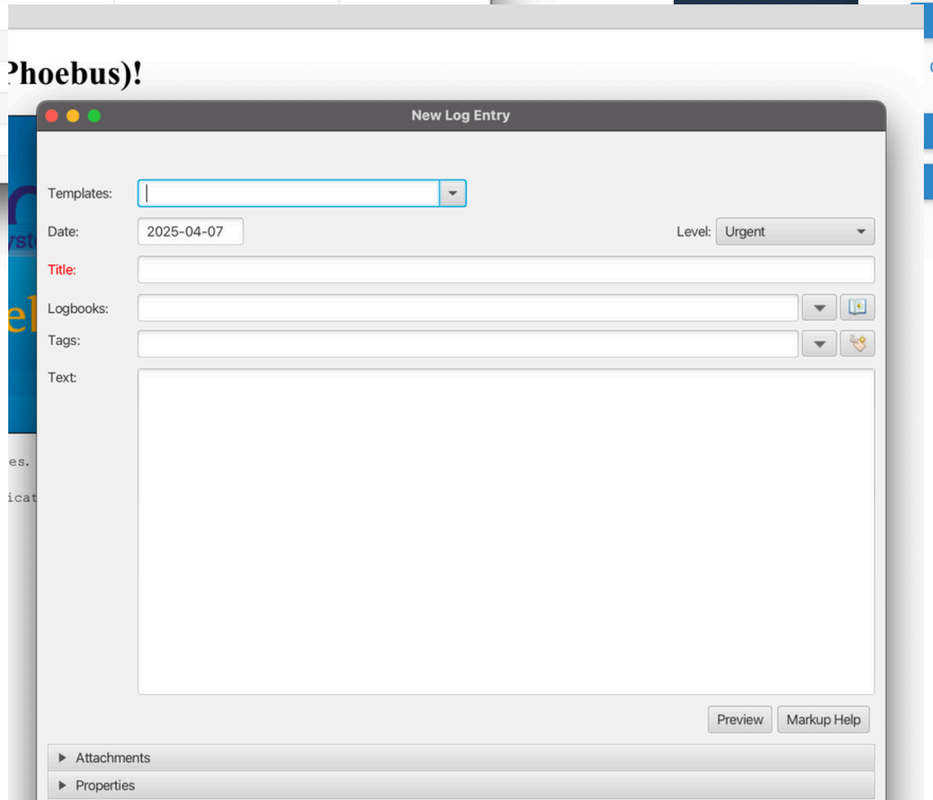
Welcome to CS-Studio (Phoebus)!

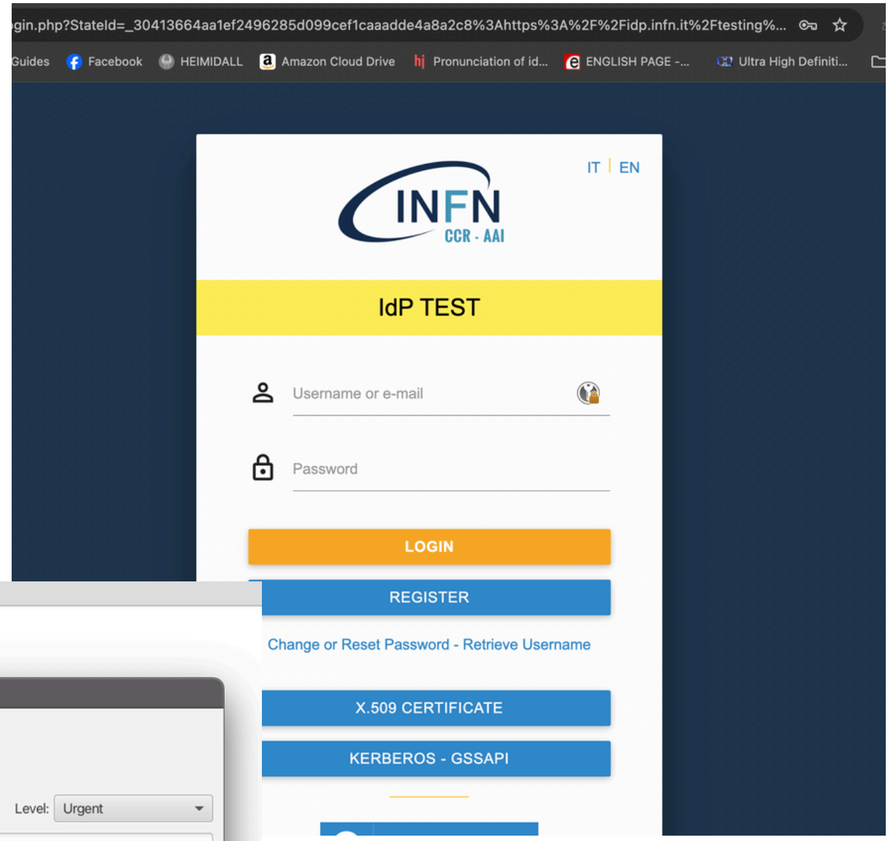


To get started, try the menu item `File`, `Top Resources`.

To install example displays, use the menu item `Applications`, `Display`, `Exa`







Pros vs Cons

“One ~~ring~~ token to rule them all.”

Single authentication for all services (Single Sign-On)

Requires an Identity Provider and Service Provider.

Create an external service in order to manage identities.

Avoid to implement login forms

Use external service to authenticate

Token expiration

Handling expiration to avoid token leakage and new login

Increase security

Credential are not exposed to individual application.

Current Development

	OAuth2 + OpenIDConnect
Phoebus Client	✓
Phoebus-Olog	✓
Olog-Web	✓
Save And Restore	⌛

✓ Done ⌛ In progress

Thanks for your attention