didmos 2.0

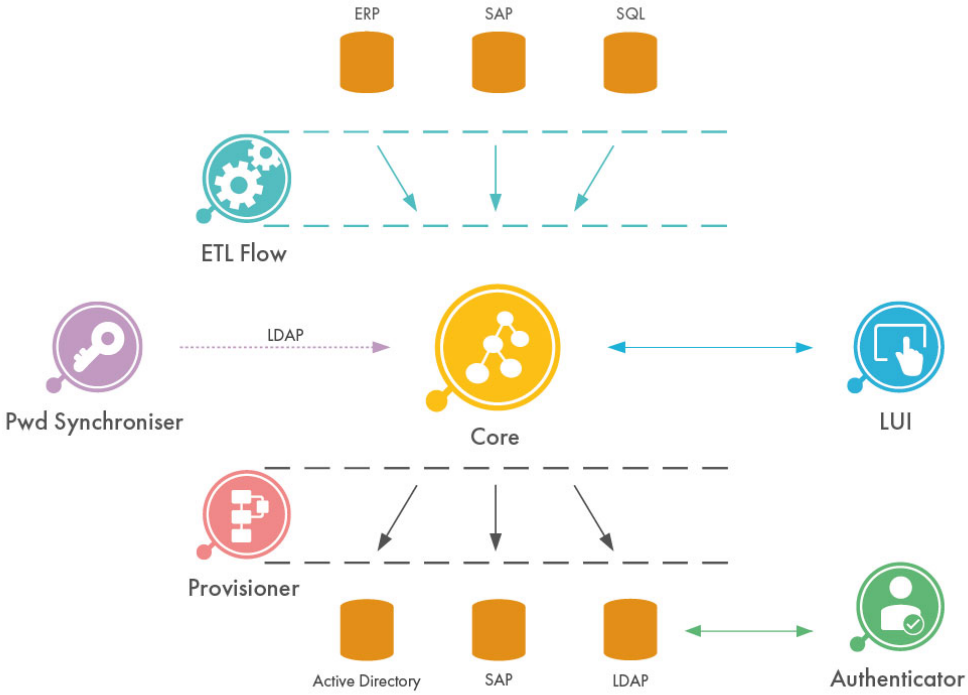
Introduction

didmos is a software solution for Identity & Access Management and consists of various different modules, that can be used individually for different tasks or in combination.

For a general overview of the software solution, please refer to the following document:

Various parts of the didmos software suite have already been updated to version 2. This version is a consequent move towards a modern micro services based software architecture for the already existing didmos modules. In addition, completely new modules are provided. This documentation covers all of these modules.

Modules



didmos V2 is made up of the following individual modules, which are illustrated above. Each module consists of a general version, which is open source and can be accessed as described below. Additionally, most modules can be extended at specific extension points to add custom functionality. A general purpose version of the frontend application (LUI) is also published as 'didmos2-demo-frontend'.

Module	Documentation	Source code	Remarks

Core	didmos2 Core	▪ https://gitlab.daasi.de/didmos2/didmos2-core (Backend APIs)
		▪ https://gitlab.daasi.de/didmos2/didmos2-openldap (Metadirectory)
			Consists of the following components:
			▪ Metadirectory (OpenLDAP)
			▪ Backend APIs consisting of different apps:
			  ▪ REST-based SCIM app
			  ▪ Policy Decision Point app
			  ▪ Task Management Engine app
			▪ For customer specific logic any number of additional apps with own web services interface can be added

LUI	didmos2 LUI	▪ https://gitlab.daasi.de/didmos2-demo/didmos2-demo-frontend (Example frontend) ▪ https://gitlab.daasi.de/didmos2/didmos2-frontend-lib (Library with common modules)	Highly customizable Frontend architecture for implementing applications communicating with the REST-APIs of Core
Authenticator	didmos2 Authenticator	▪ https://gitlab.daasi.de/didmos2/didmos2-auth (SSO platform) ▪ https://gitlab.daasi.de/didmos2/didmos2-mongodb (MongoDB storage layer)	Consists of the following components: ▪ SSO proxy (SAML and OIDC) based on Satosa from the IdentityPython project ▪ uses MongoDB for short lived tokens
Provisioner	didmos2 Provisioner	▪ https://gitlab.daasi.de/didmos2/didmos2-ra (RA) ▪ https://gitlab.daasi.de/didmos2/didmos2-rabbitmq-worker (Worker) ▪ https://gitlab.daasi.de/didmos2/didmos2-connector-scim2 (Generic SCIM connector) ▪ https://gitlab.daasi.de/didmos2/didmos2-worker-scim2-response (Integration of SCIM connector for writing Provisioner responses to didmos Core)	Consists of the following components: ▪ Requesting Authority (RA) ▪ RabbitMQ (Queue system, see Dockerhub) ▪ Worker-Nodes with ICF connectors for various target systems
ETL Flow			Not yet updated to didmos V2, didmos V1 is compatible with didmos V2

Pwd Synchronizer			Not yet updated to didmos V2, didmos V1 is compatible with didmos V2

How to get started

In addition to the source code repositories, there is also a dedicated 'compose'-repository which contains a Docker-based development environment, build pipelines and deployment instructions. The general purpose version of didmos is called didmos2-demo and its compose-repository is accessible here: https://gitlab.daasi.de/didmos2-demo/didmos2-demo-compose

At that location you can find details on how to setup a development environment, which currently is only fully documented for the Fedora operating system. The repository can also be used to setup a local demo environment based on pre-build docker images.

The most recent version of didmos2-demo is v2.2.0: https://gitlab.daasi.de/didmos2-demo/didmos2-demo-compose/-/tree/v2.2.0

Below you can find a short summary of how to setup a local demo environment. For more details and deployment scenarios, please refer to the full README: https://gitlab.daasi.de/didmos2-demo/didmos2-demo-compose/-/blob/v2.2.0/README.md

Setup of a local demo environment

```
# !!!
# Make sure to understand the "General requirements" section in https://gitlab.daasi.de/didmos2-demo/didmos2-demo-compose/-/blob/v2.2.0/README.md#general-requirements
# !!!

# Clone didmos2-demo-compose repository and change diretcory
git clone -b v2.2.0 https://gitlab.daasi.de/didmos2-demo/didmos2-demo-compose.git
cd didmos2-demo-compose

# Run bootstrap script with parameters for release-branches, external Gitlab and docker environment only
make bootstrap ENV=release GIT_PROFILE=external DEPLOY=dockerOnly

# Run docker containers
make up

# !!!
# Make sure to whitelist/accept SSL certificates for
# - https://auth.daasi.devel
# - https://backend.daasi.devel
# - https://frontend.daasi.devel
# And then finally access https://frontend.daasi.devel. Login is possible with user superadmin and password secret
# !!!
```

Operations

Documentation for production deployment and operations is available here: Operations