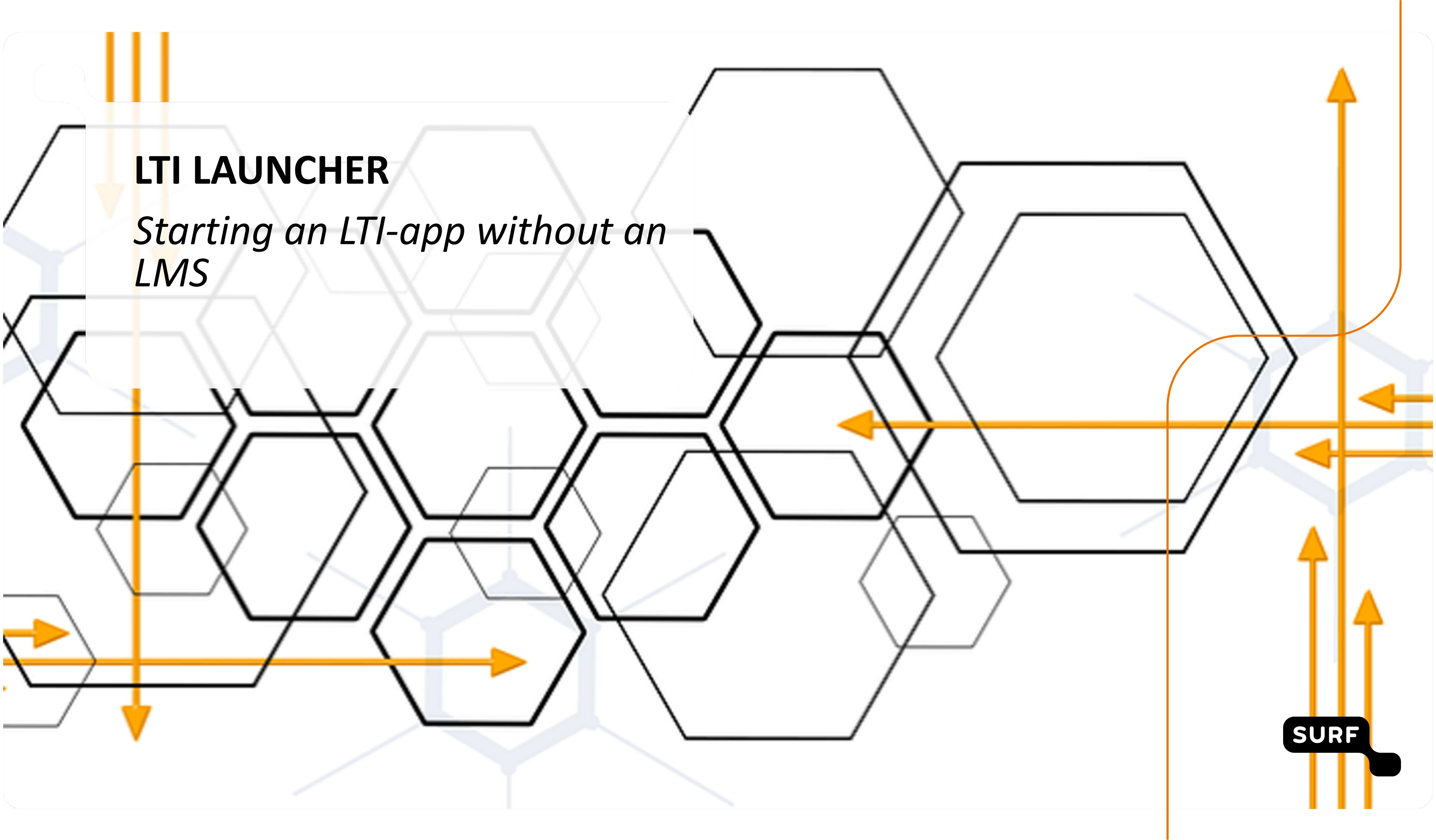
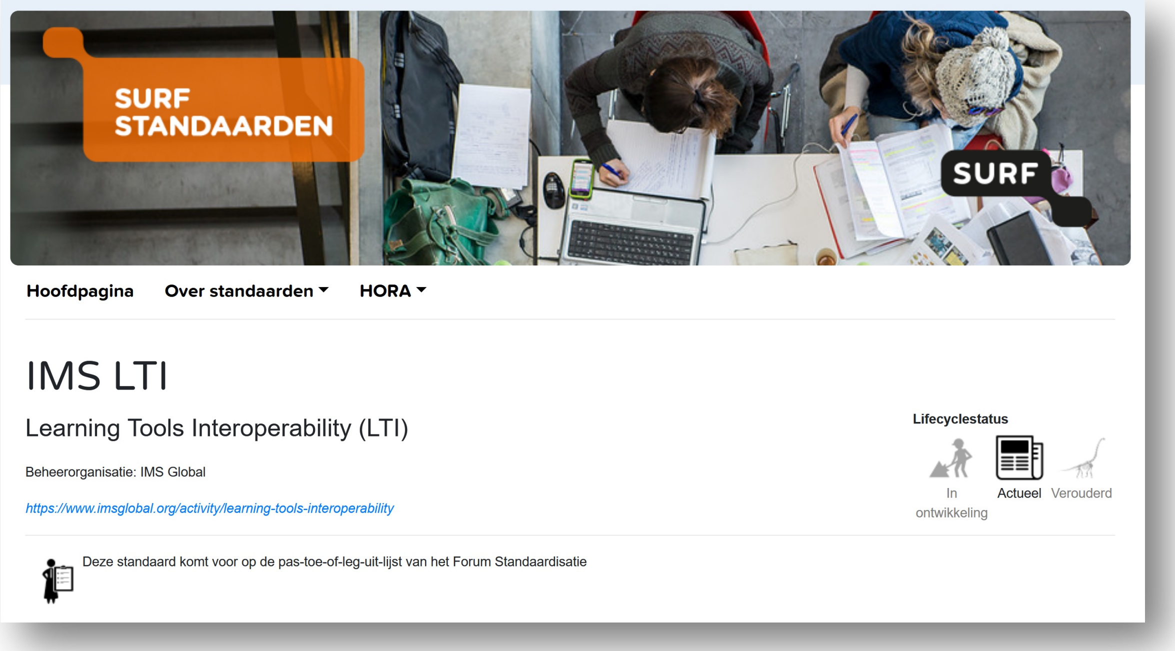
LTI LAUNCHER

Starting an LTI-app without an LMS



SURF

Standards



Hoofdpagina Over standaarden ▾ HORA ▾

IMS LTI

Learning Tools Interoperability (LTI)

Beheerorganisatie: IMS Global

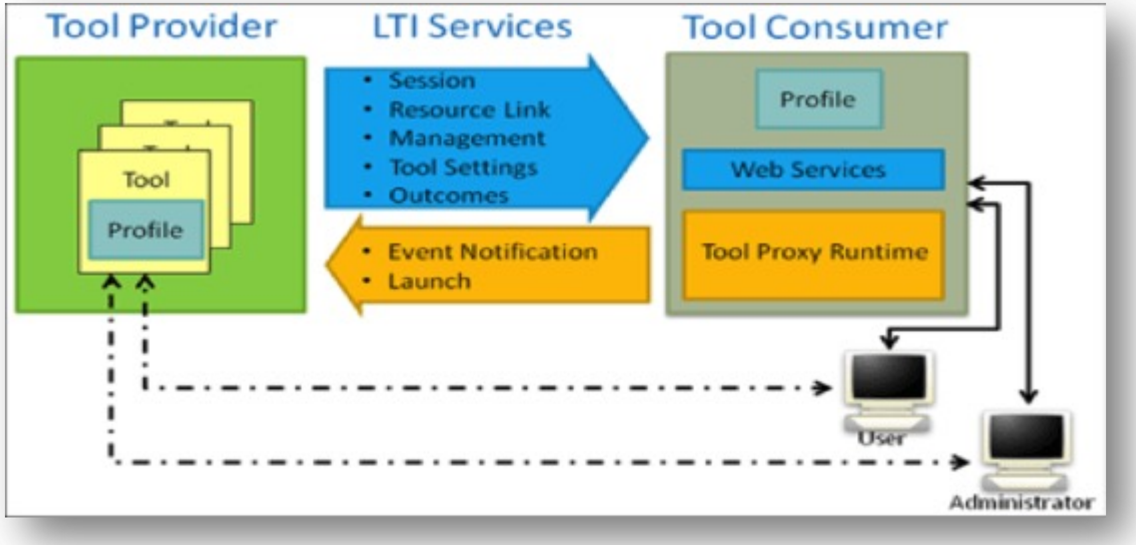
https://www.imsglobal.org/activity/learning-tools-interoperability

Lifecyclestatus

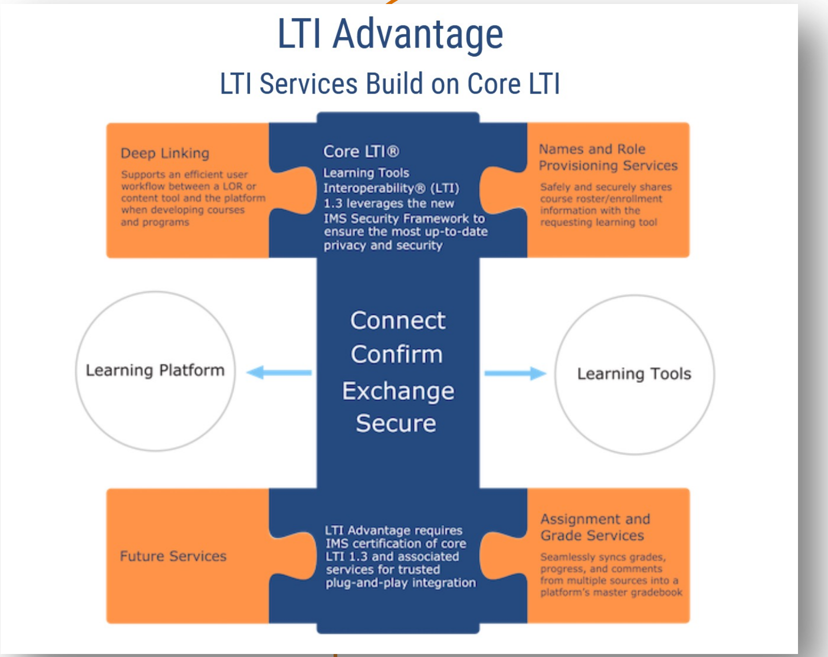
In ontwikkeling Actueel Verouderd

Deze standaard komt voor op de pas-toe-of-leg-uit-lijst van het Forum Standaardisatie

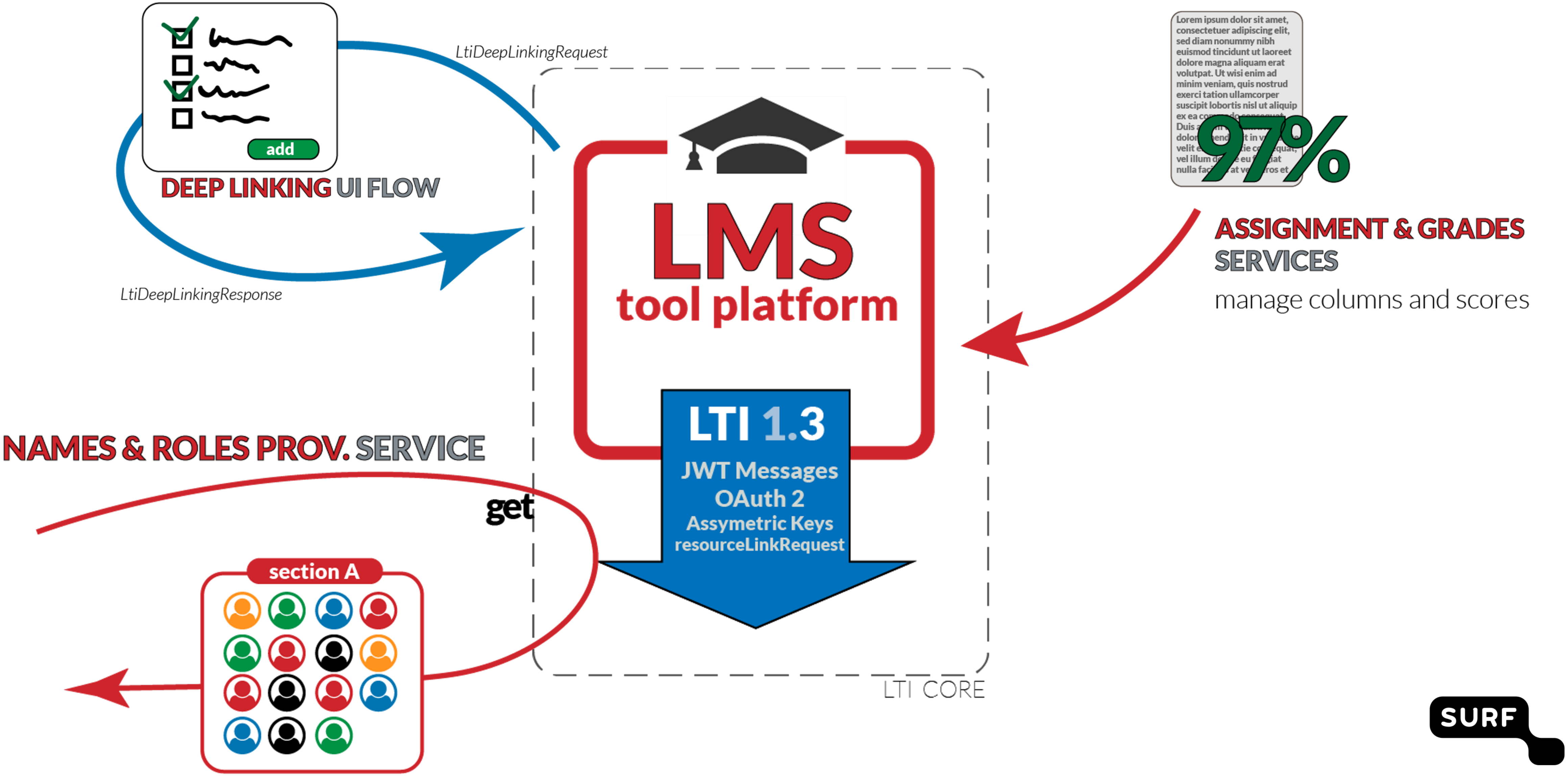
FROM 1.2 (OAUTH 1.0)



TO 1.3 (OAUTH 2.0 AND OPENID CONNECT)


LTI Advantage
LTI Services Build on Core LTI
Deep Linking
Supports an efficient user workflow between a LOR or content tool and the platform when developing courses and programs
Core LTI®
Learning Tools Interoperability® (LTI) 1.3 leverages the new IMS Security Framework to ensure the most up-to-date privacy and security
Names and Role Provisioning Services
Safely and securely shares course roster/enrollment information with the requesting learning tool
Learning Platform
Connect
Confirm
Exchange
Secure
Learning Tools
Future Services
LTI Advantage requires IMS certification of core LTI 1.3 and associated services for trusted plug-and-play integration
Assignment and Grade Services
Seamlessly syncs grades, progress, and comments from multiple sources into a platform's master gradebook

LTI Advantage = LTI 1.3 + 3 services



DEEP LINKING UI FLOW

LtiDeepLinkingRequest

LtiDeepLinkingResponse

LMS

tool platform

LTI 1.3

JWT Messages
OAuth 2
Assymetric Keys
resourceLinkRequest

LTI CORE

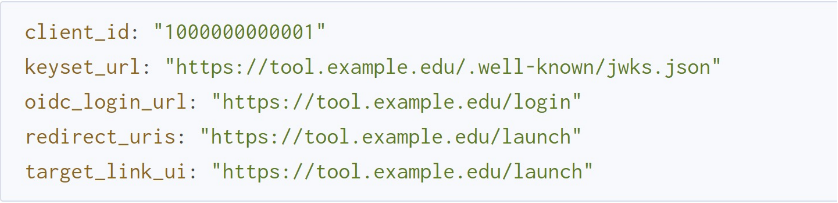
ASSIGNMENT & GRADES SERVICES

manage columns and scores

NAMES & ROLES PROV. SERVICE

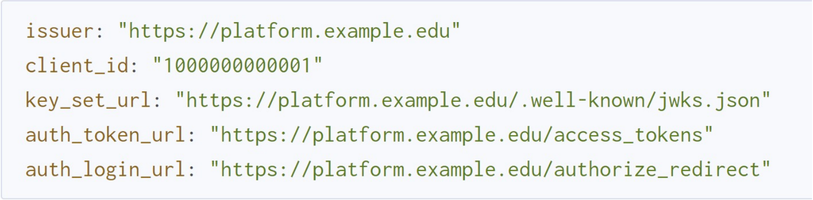
SURF

Example tool details registered with the platform:

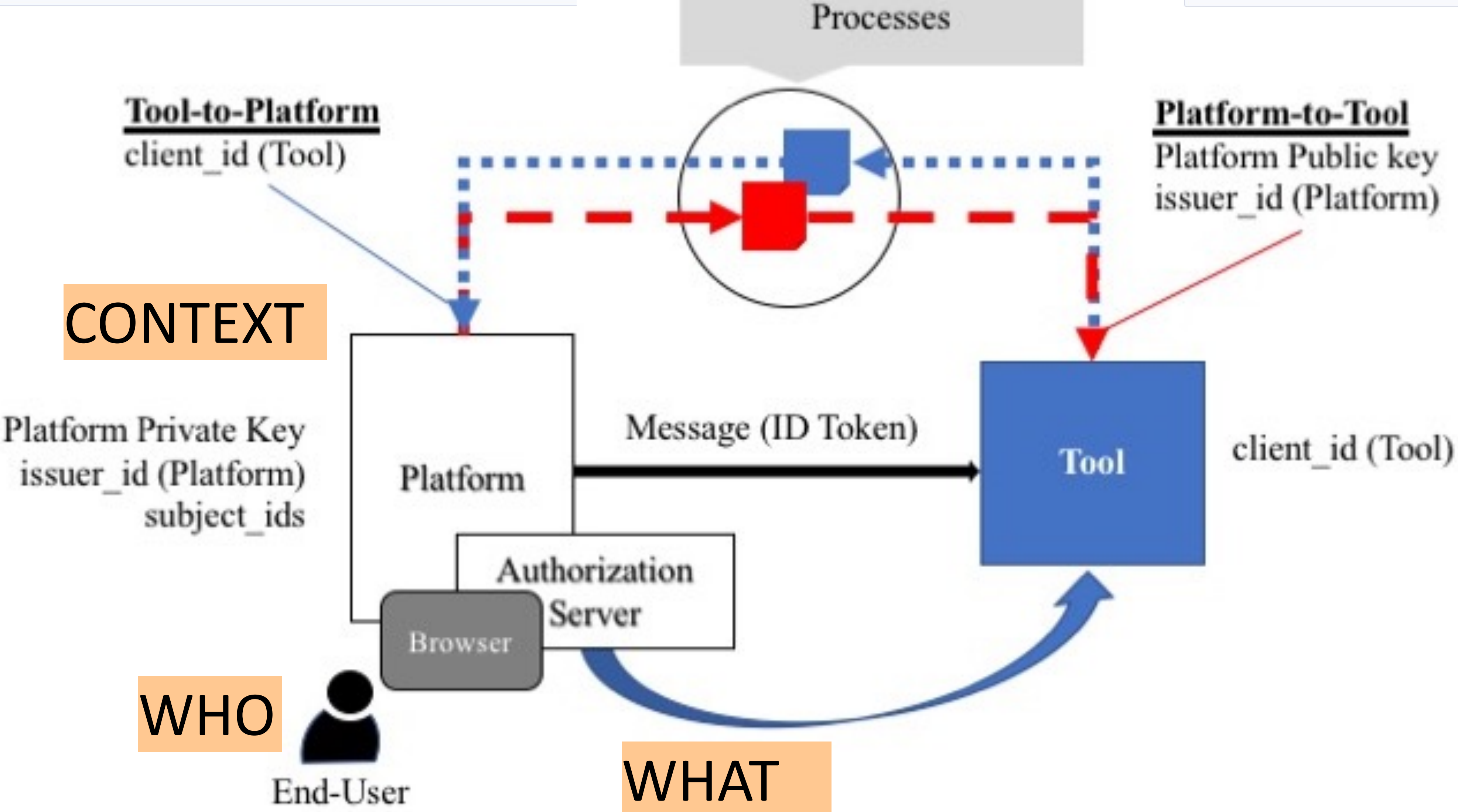


```
client_id: "1000000000001"
keyset_url: "https://tool.example.edu/.well-known/jwks.json"
oidc_login_url: "https://tool.example.edu/login"
redirect_uris: "https://tool.example.edu/launch"
target_link_ui: "https://tool.example.edu/launch"
```

Example platform details registered with the tool:



```
issuer: "https://platform.example.edu"
client_id: "1000000000001"
key_set_url: "https://platform.example.edu/.well-known/jwks.json"
auth_token_url: "https://platform.example.edu/access_tokens"
auth_login_url: "https://platform.example.edu/authorize_redirect"
```



CONTEXT

WHO

WHAT

Why a separate LTI-Launcher (1)

- Launch without an LMS
  - **Decouple** LTI tools from the LMS
  - Enable **more entry points** for a tool
  - Provide **extra focus** on a tool

- Add context manually
  - Context is not always required or can be fixed
  - Pass additional context to the launcher (using JWT)

Why a separate LTI-Launcher (2)

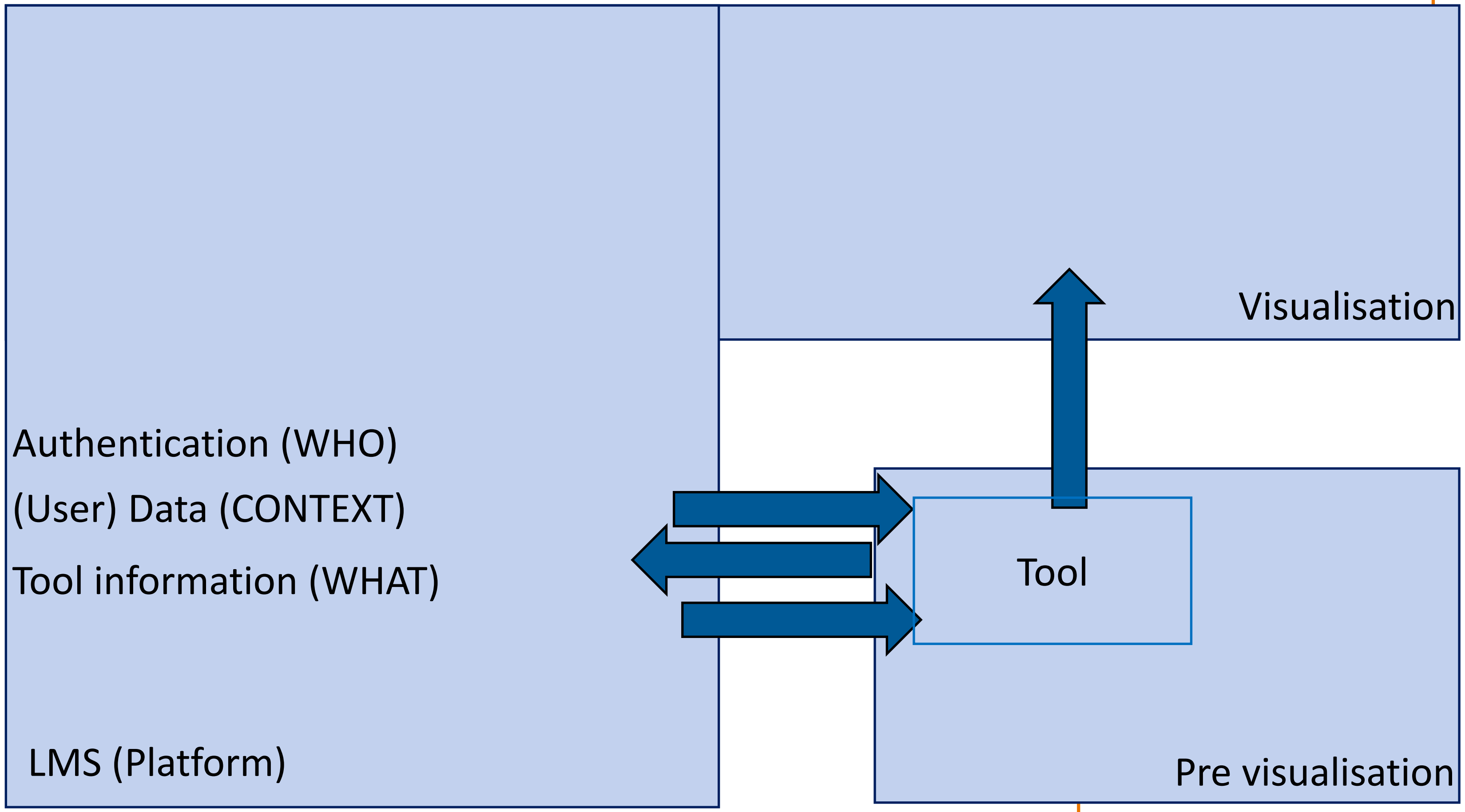
- Version agnostic
  - Platforms don’t care about used LTI version
  - New versions of LTI standard can be fixed in a single location

- For **easy** testing
  - Test LTI capabilities of tools and **using underlying open standards** without using a full LMS

How does the LTI launcher work

SURF

Overview



Authentication (WHO)

(User) Data (CONTEXT)

Tool information (WHAT)

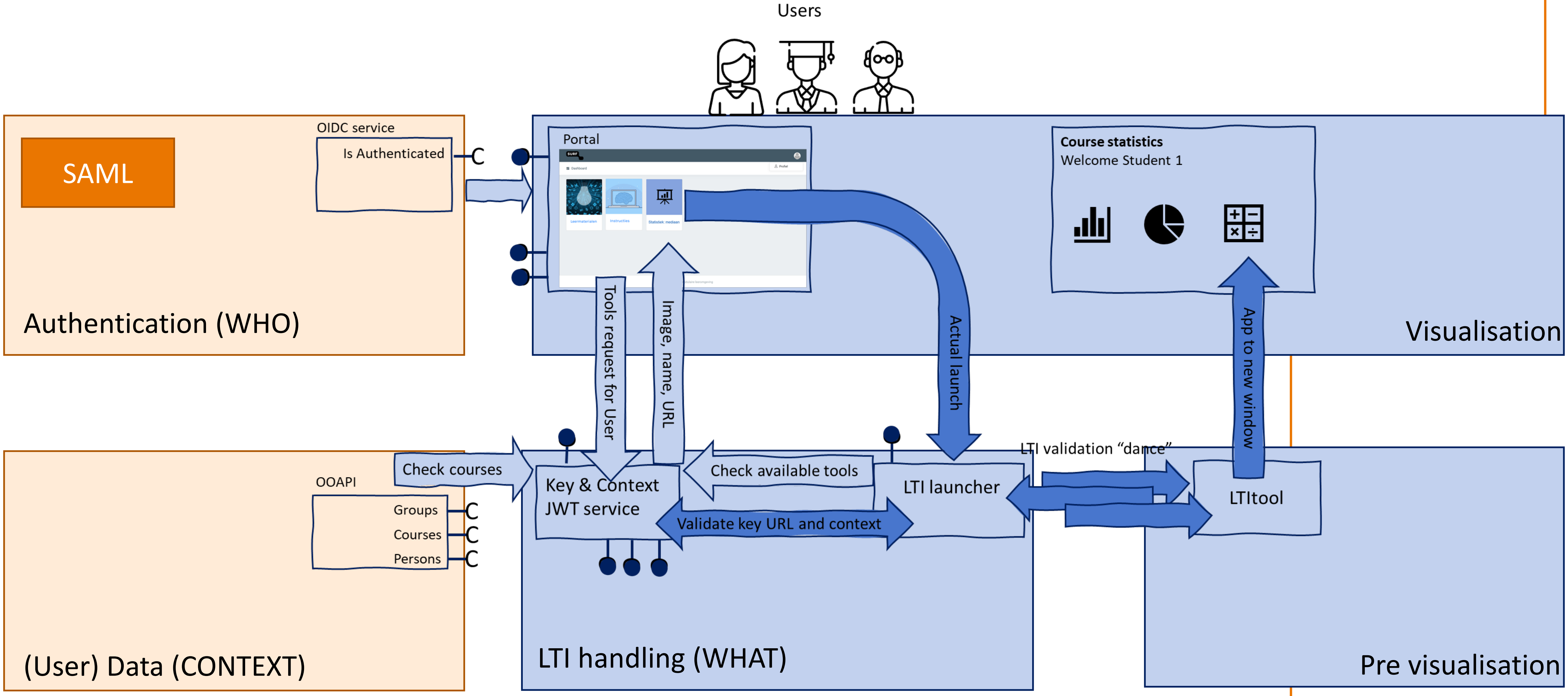
LMS (Platform)

Tool

Visualisation

Pre visualisation

Overview



Users

Authentication (WHO)

SAML

OIDC service

Is Authenticated

(User) Data (CONTEXT)

OOAPI

Groups

Courses

Persons

Check courses

Visualisation

Portal

Course statistics

Welcome Student 1

Tools request for User

Image, name, URL

Actual launch

App to new window

LTI handling (WHAT)

Key & Context JWT service

Check available tools

Validate key URL and context

LTI launcher

LTI validation “dance”

LTItool

Pre visualisation

How does it work (information gathering)

- **Manage tools**
  - The launcher offers an administrative interface to set-up tools including security credentials.

- **Get the WHO**

- The Launcher is directly linked to a Single Sign On (SSO) service.
  - Only supports OpenID Connect for now.

How does it work (information gathering)

- ## Get the WHAT

- A user ‘selects’ the tool by going to a specific URL (or click a link):

- ## Get the CONTEXT

- The generator of the link can add context by adding a JWT to the URL:

https://lti-launcher.com/launch/**<tool_id>** ? **context=xxx.xxx.xxx**

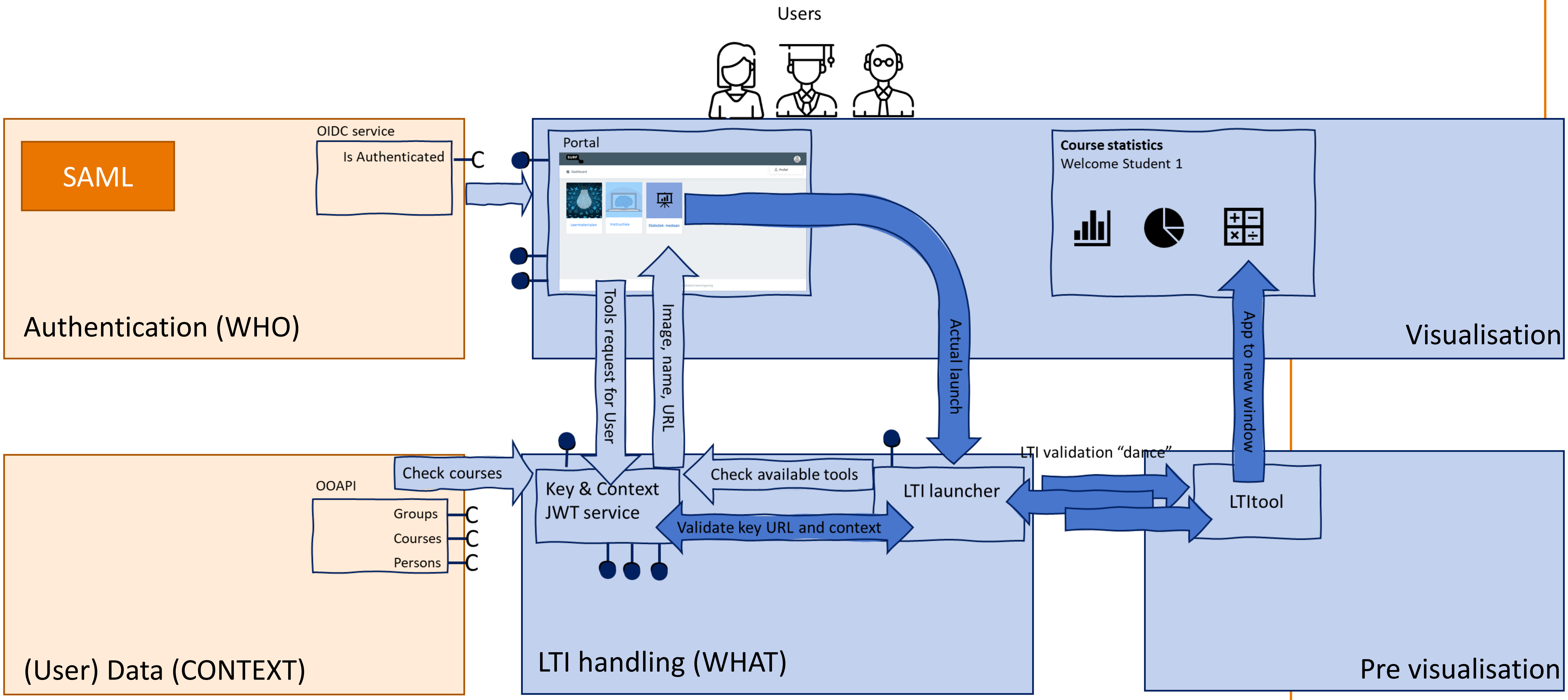
How does it work (the LTI 1.3 dance)

- **Launching the LTI “dance”**

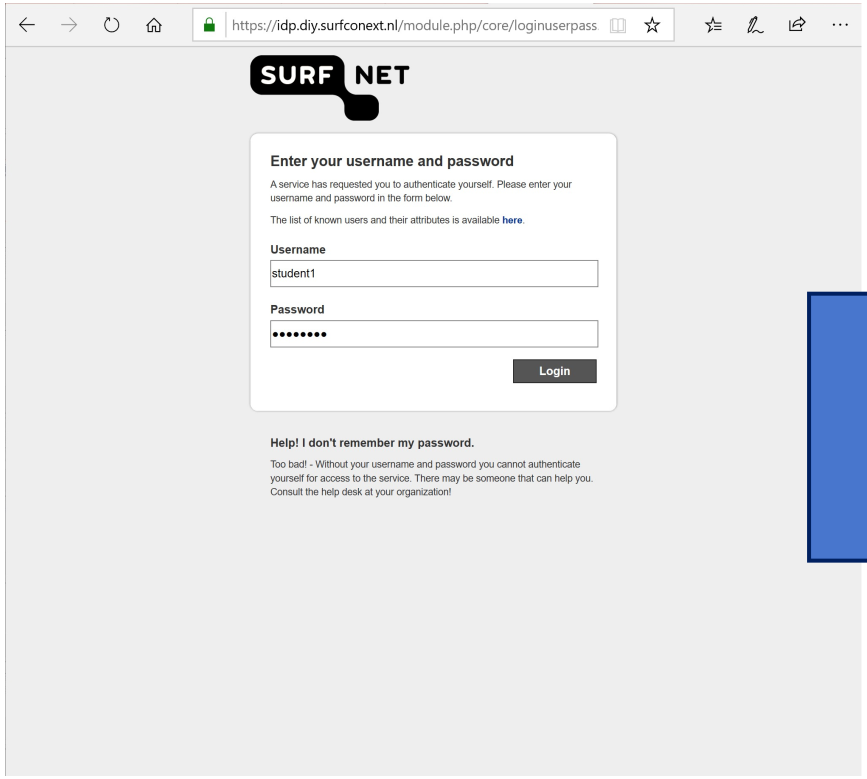
1. User navigates to: https://lti-launcher.com/launch/<tool_id>?context=xxx.xxx.xxx
2. Launcher redirects user to SSO service to get user information
3. User logs in if needed and gets redirected to the launcher
4. Launcher starts Open ID Connect flow with LTI 1.3 tool
5. User gets redirected back to the launcher
6. Launcher performs the final launch

Visuals

Overview


Users
OIDC service
SAML
Is Authenticated
Authentication (WHO)
Portal
Tools request for User
Image, name, URL
Actual launch
Course statistics
Welcome Student 1
App to new window
Visualisation
OOAPI
Check courses
Groups
Courses
Persons
(User) Data (CONTEXT)
Key & Context
JWT service
Check available tools
LTI launcher
Validate key URL and context
LTI validation "dance"
LTItool
LTI handling (WHAT)
Pre visualisation

Student opens portal site and logs in



Enter your username and password

A service has requested you to authenticate yourself. Please enter your username and password in the form below.

The list of known users and their attributes is available **here**.

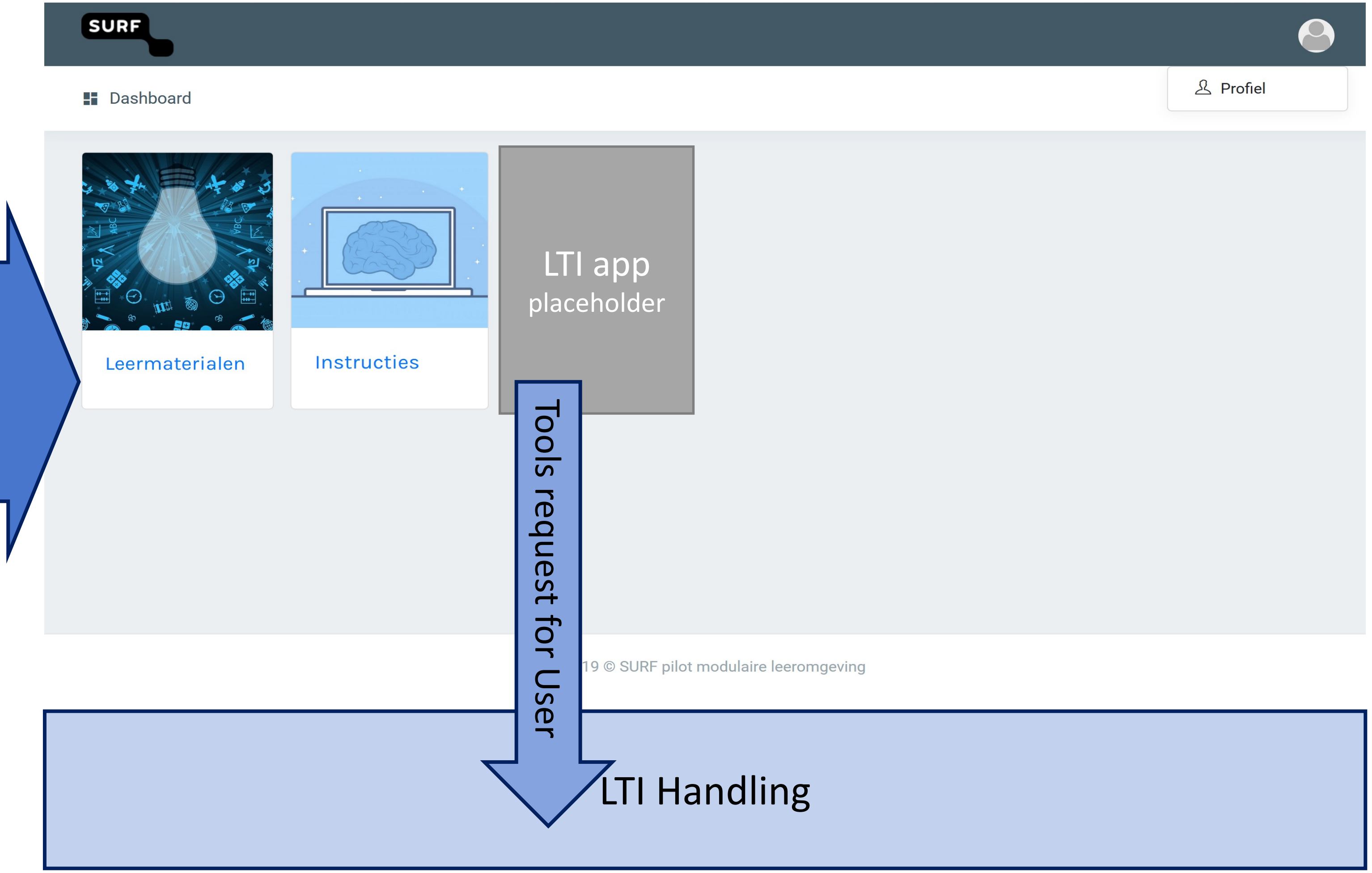
Username: student1

Password: ••••••••

Login

Help! I don't remember my password.

Too bad! - Without your username and password you cannot authenticate yourself for access to the service. There may be someone that can help you. Consult the help desk at your organization!

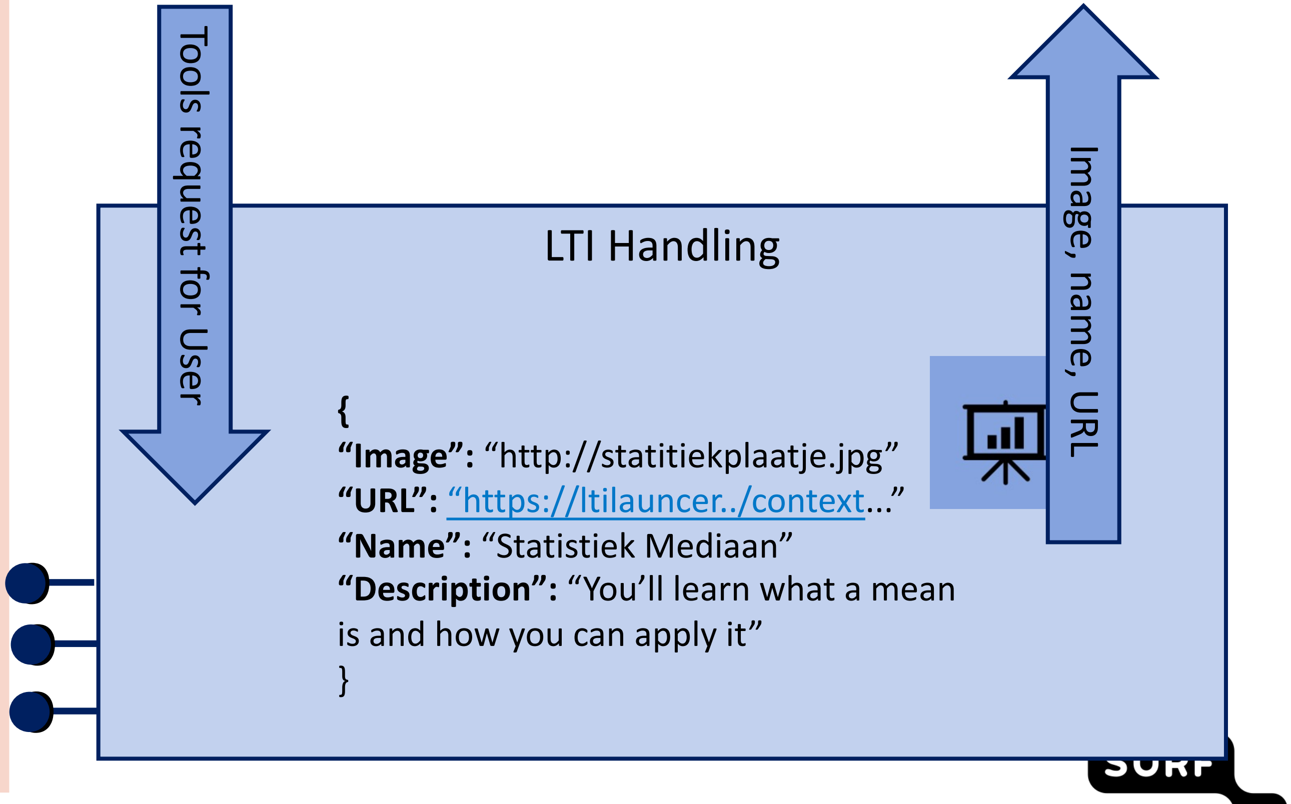


Generate the tile for the portal

JWT Service:

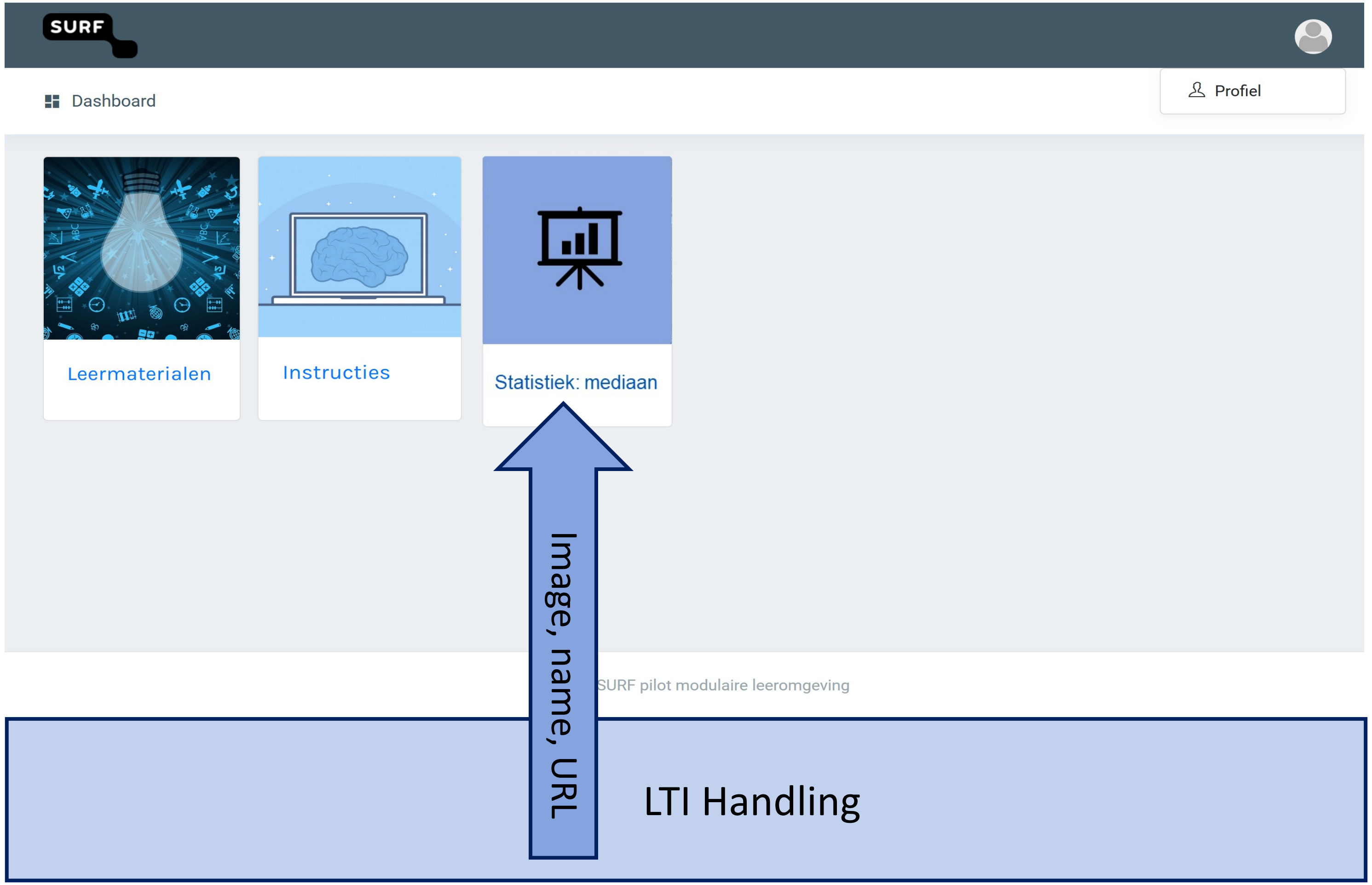
- Search OOAPI
  - Person (find out current role)
  - Courses of student (which courses is student currently active in)
  - LTI resources of sed courses

  - Generate context for each resource
  - Search metadata for LTI resources in LTI launcher
  - Create package for portal:
    - Combine context and launcher URL
    - Image of tool
    - Name of tool
    - Description of tool



{
"Image": "http://statitiekplaatje.jpg"
"URL": "https://ltilauncer../context..."
"Name": "Statistiek Mediaan"
"Description": "You'll learn what a mean is and how you can apply it"
}

Portal is generated based on the LTI tools with context that are available to the student

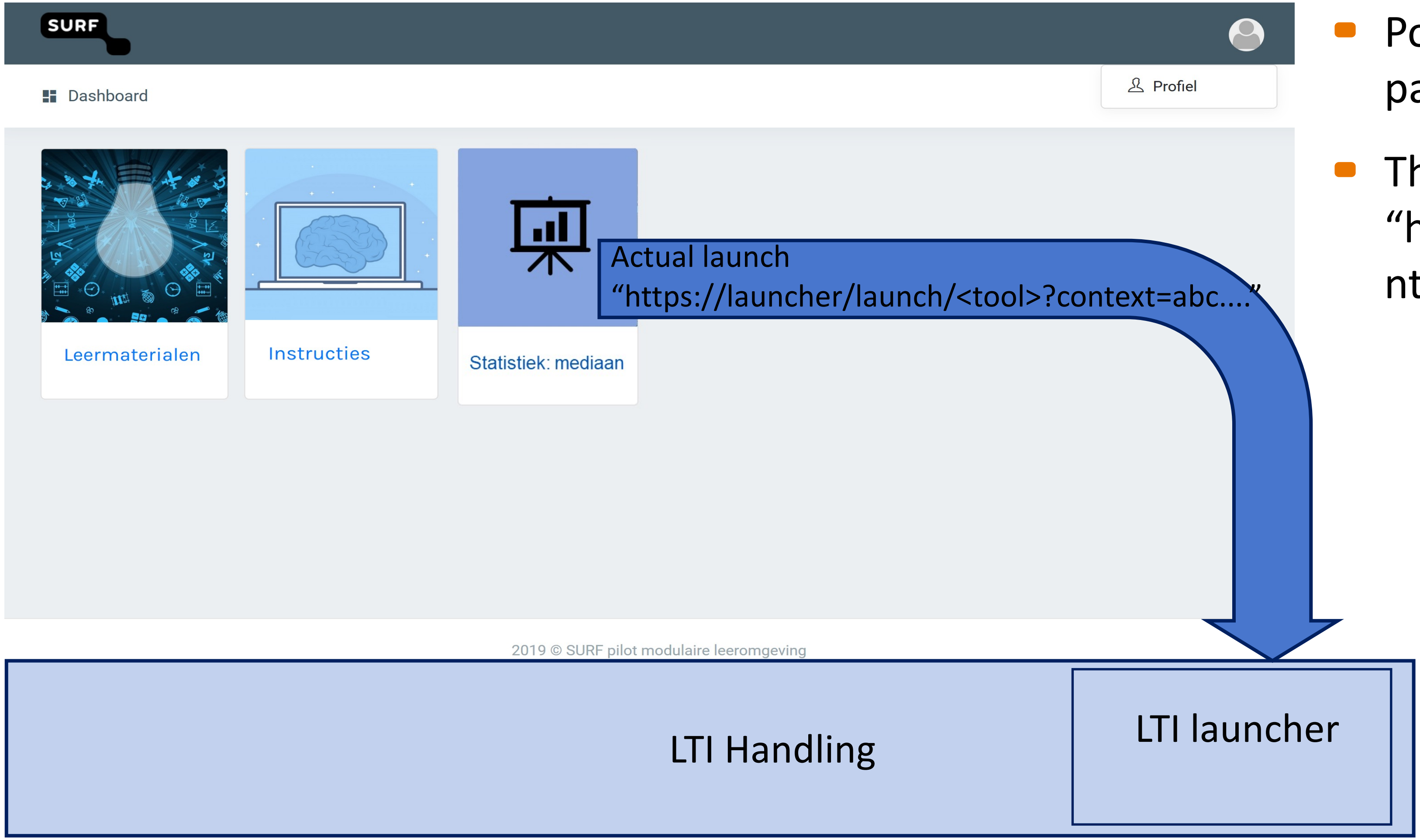


Image, name, URL

LTI Handling

- Portal generates tile based on information provided by JWT service

Student launches LTI tool



Actual launch
“https://launcher/launch/<tool>?context=abc....”

LTI Handling

LTI launcher

- Portal redirects student to new page that opens LTI launcher URL.
- This is a URL get: “https://launcher/launch/<tool>?context=abc....”

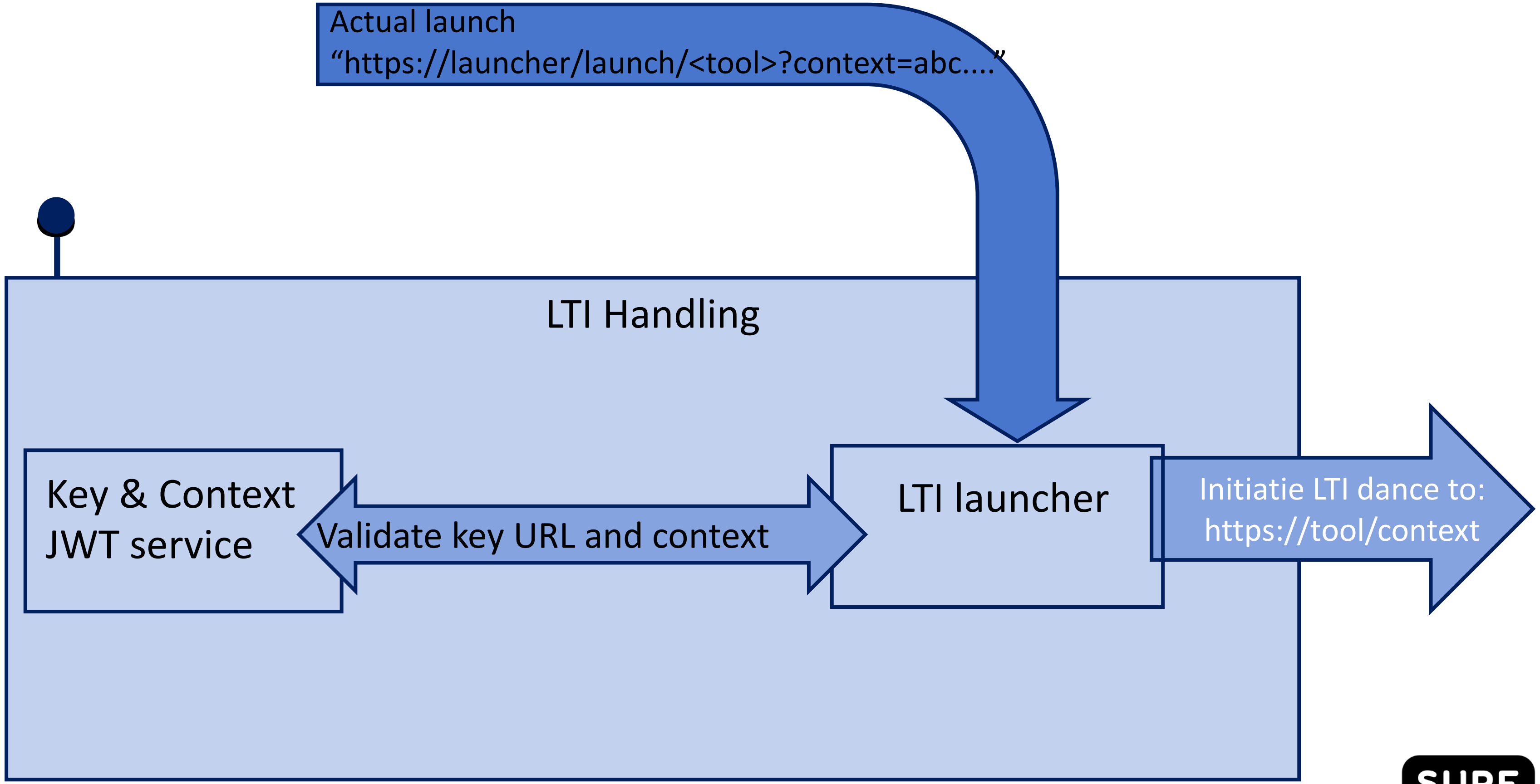
The LTI launcher handles the launch call

LTI launcher

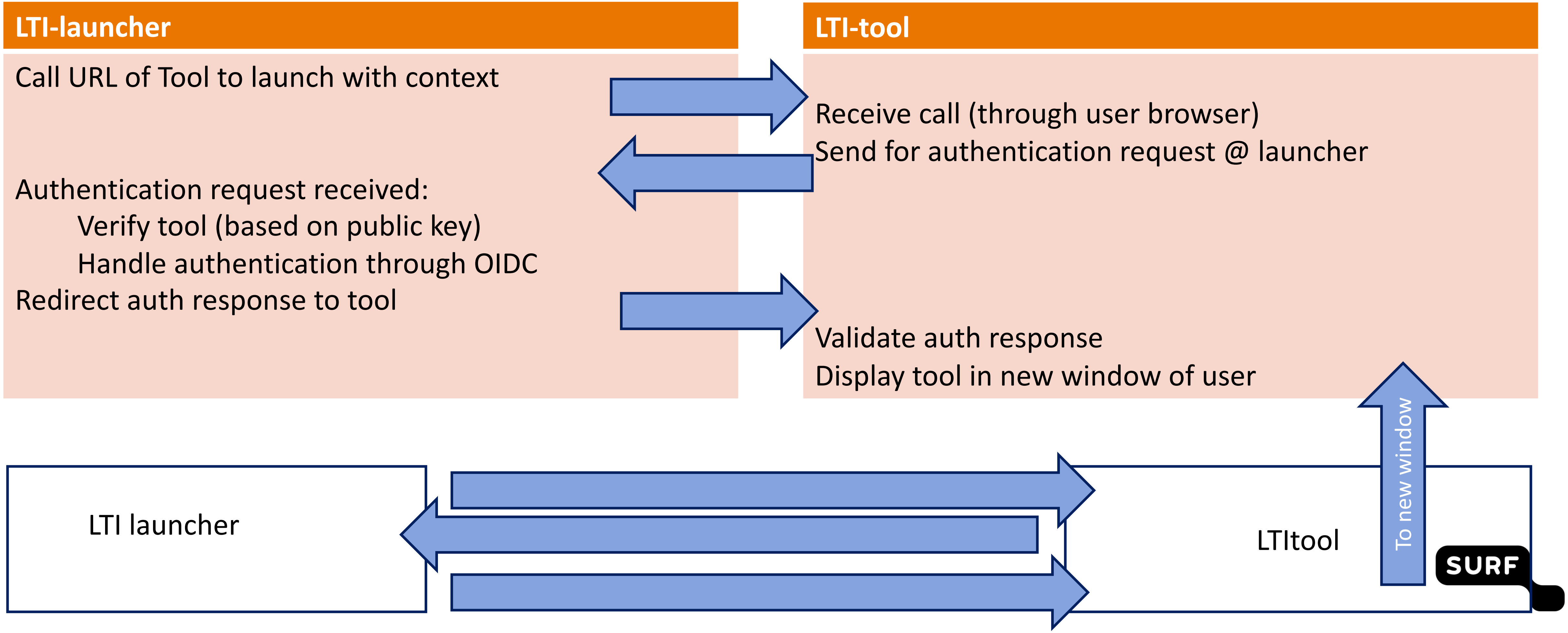
Receive intitial call:

- Check / validate context and key @ WJT service
- Check OIDC

- Forward call to LTI component associated with tool number in URL, initiating “LTI dance”
- From here on LTI tool URL is used in sted of launcher



“LTI DANCE”



LTI-launcher

Call URL of Tool to launch with context

Authentication request received:
- Verify tool (based on public key)
- Handle authentication through OIDC

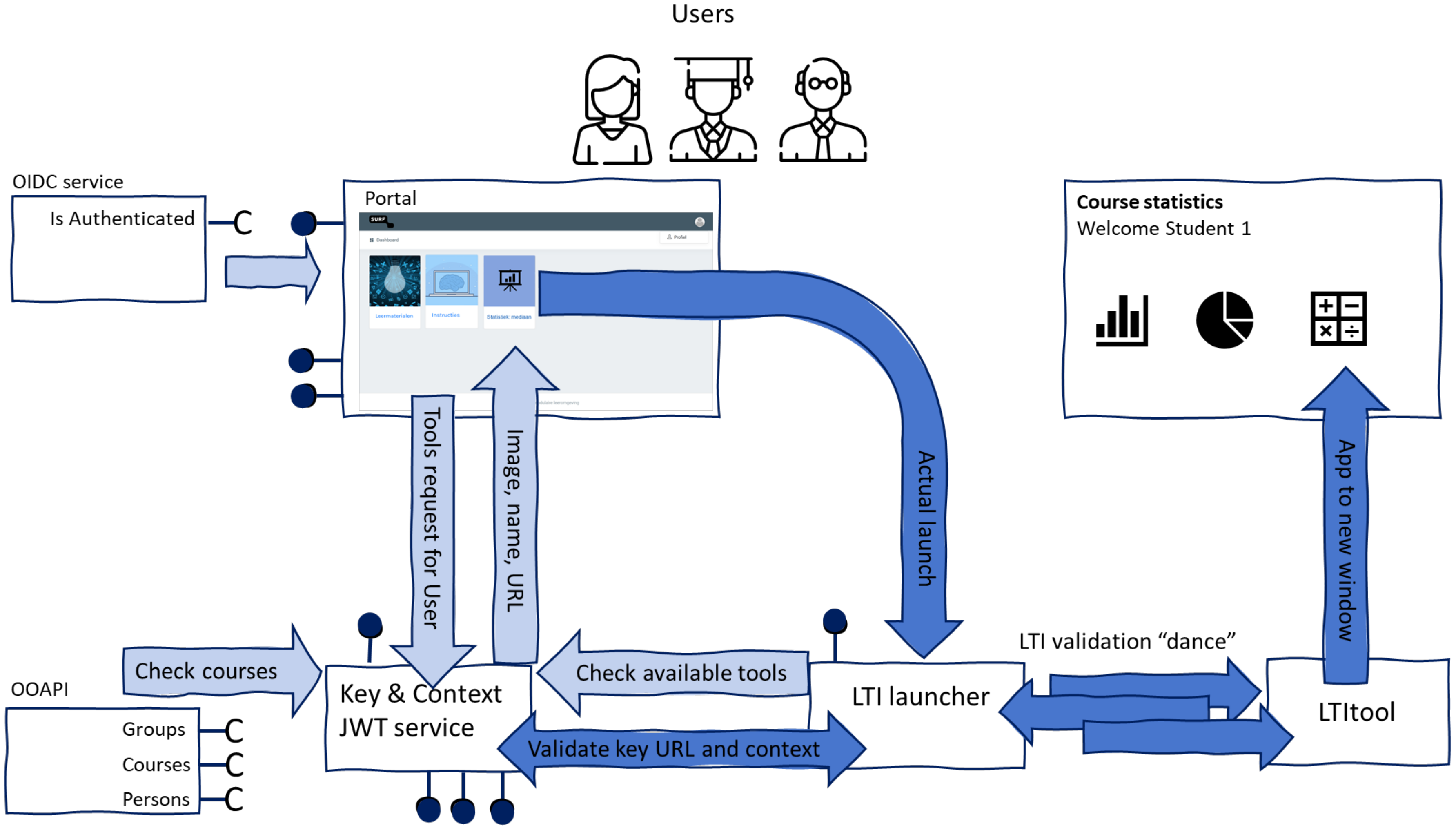
Redirect auth response to tool

LTI-tool

Receive call (through user browser)
Send for authentication request @ launcher

Validate auth response
Display tool in new window of user

The whole LTI journey for a student

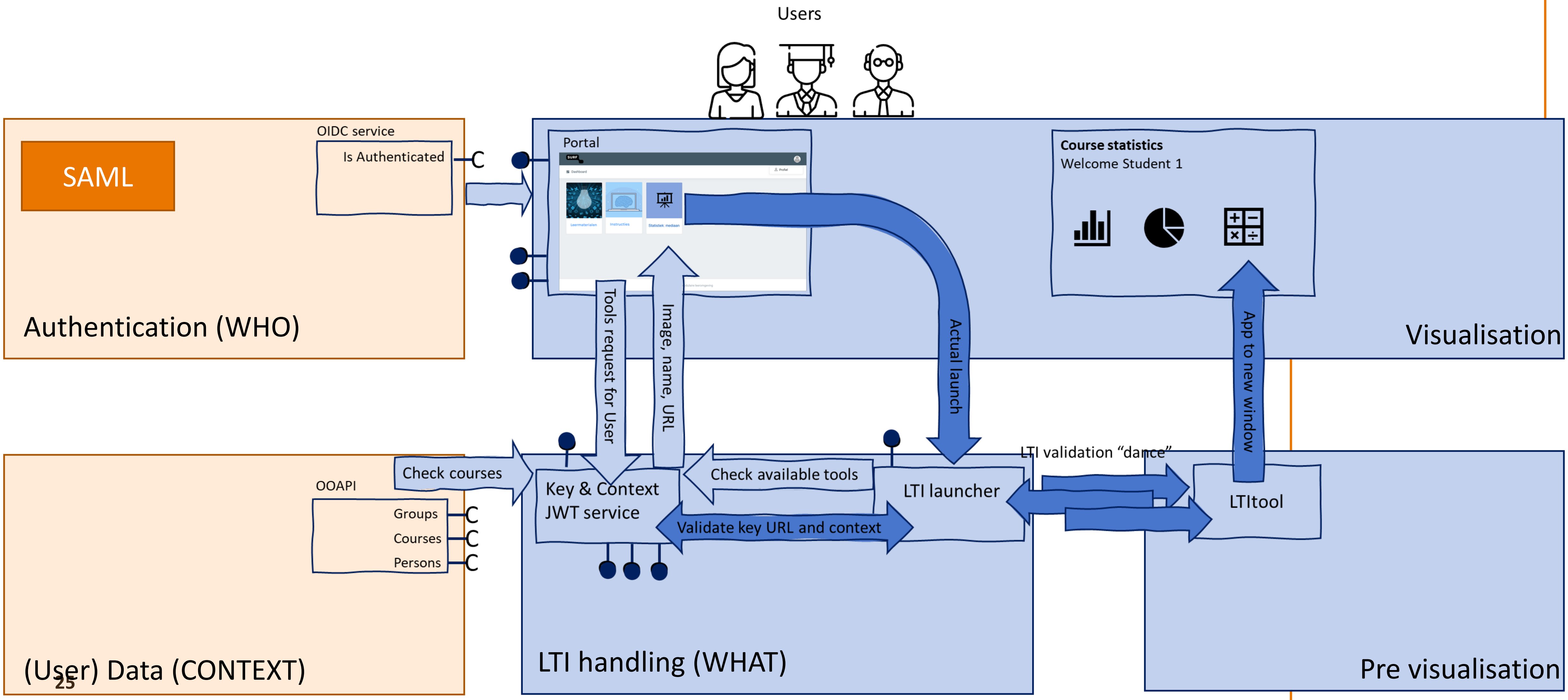


Tool demo

- Demo can be found here:
-
- https://www.imsglobal.org/sites/default/files/EUsummit2020/lti_laucher_v3.mp4

Recommendations for Future Work

Overview



Users

Portal

Course statistics

Welcome Student 1

Visualisation

OIDC service

Is Authenticated

SAML

Authentication (WHO)

Tools request for User

Image, name, URL

Actual launch

App to new window

LTI validation “dance”

OOAPI

Groups

Courses

Persons

Check courses

Key & Context JWT service

Check available tools

LTI launcher

Validate key URL and context

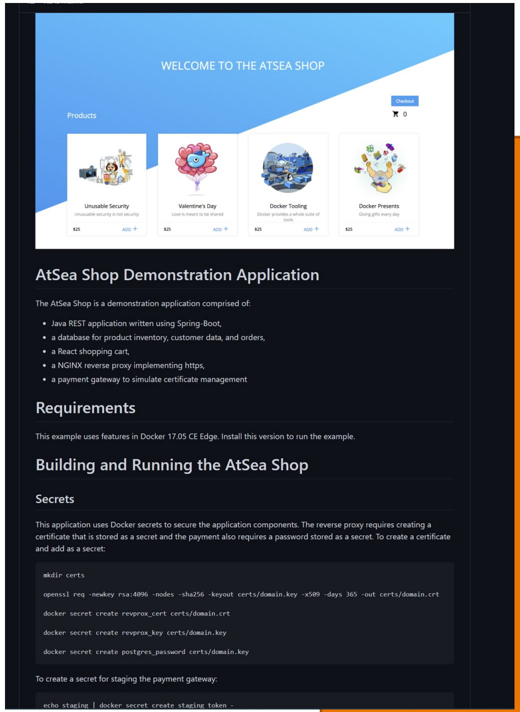
LTItool

(User) Data (CONTEXT)

LTI handling (WHAT)

Pre visualisation

Packaging comparable to AtSea



AtSea Shop Demonstration Application

The AtSea Shop is a demonstration application comprised of:

- Java REST application written using Spring-Boot,
- a database for product inventory, customer data, and orders,
- a React shopping cart,
- a NGINX reverse proxy implementing https,
- a payment gateway to simulate certificate management

Requirements

This example uses features in Docker 17.05 CE Edge. Install this version to run the example.

Building and Running the AtSea Shop

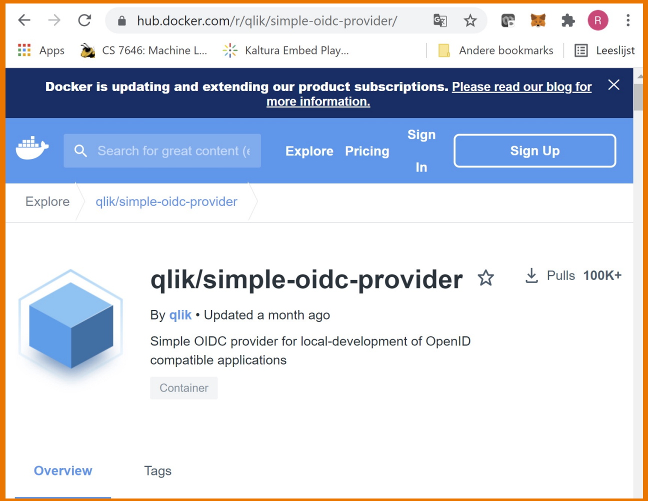
Secrets

This application uses Docker secrets to secure the application components. The reverse proxy requires creating a certificate that is stored as a secret and the payment also requires a password stored as a secret. To create a certificate and add as a secret:

```
mkdir certs

openssl req -newkey rsa:4096 -nodes -sha256 -keyout certs/domain.key -x509 -days 365 -out certs/domain.crt

docker secret create revprox_cert certs/domain.crt

docker secret create revprox_key certs/domain.key

docker secret create postgres_password certs/domain.key
```

To create a secret for staging the payment gateway:

```
echo staging | docker secret create staging token -
```



Docker is updating and extending our product subscriptions. Please read our blog for more information.

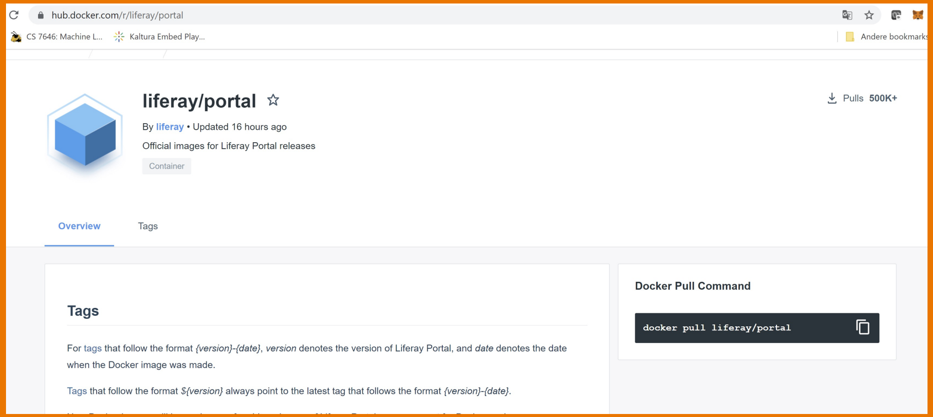
qlik/simple-oidc-provider

Pulls 100K+

By qlik • Updated a month ago

Simple OIDC provider for local-development of OpenID compatible applications

Separate OIDC (WHO)



liferay/portal

Pulls 500K+

By liferay • Updated 16 hours ago

Official images for Liferay Portal releases

Tags

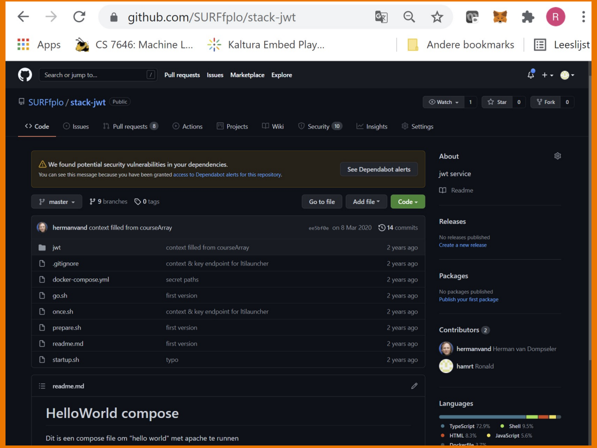
For tags that follow the format *{version}-{date}*, *version* denotes the version of Liferay Portal, and *date* denotes the date when the Docker image was made.

Tags that follow the format *${version}* always point to the latest tag that follows the format *{version}-{date}*.

Docker Pull Command

```
docker pull liferay/portal
```

Separate portal (VISUALISATION)



We found potential security vulnerabilities in your dependencies.
You can see this message because you have been granted access to Dependabot alerts for this repository.

hermanvand context filled from courseArray ee5bf0e on 8 Mar 2020 14 commits

jwt	context filled from courseArray	2 years ago
.gitignore	context & key endpoint for ltilauncher	2 years ago
docker-compose.yml	secret paths	2 years ago
go.sh	first version	2 years ago
once.sh	context & key endpoint for ltilauncher	2 years ago
prepare.sh	first version	2 years ago
readme.md	first version	2 years ago
startup.sh	typo	2 years ago

readme.md

HelloWorld compose

Dit is een compose file om "hello world" met apache te runnen

About

jwt service

Releases

No releases published

Packages

No packages published

Contributors 2

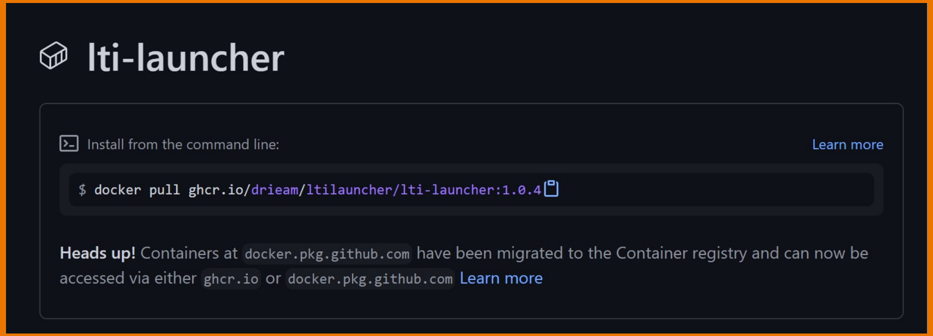
hermanvand Herman van Dompseler

hamrt Ronald

Languages

TypeScript 72.9% Shell 9.5% HTML 8.3% JavaScript 5.6% Dockerfile 3.7%

Separate JWT service (WHAT+CONTEXT)



lti-launcher

Install from the command line:

Learn more

```
$ docker pull ghcr.io/drieam/ltilauncher/lti-launcher:1.0.4
```

Heads up! Containers at `docker.pkg.github.com` have been migrated to the Container registry and can now be accessed via either `ghcr.io` or `docker.pkg.github.com` Learn more

Separate LTI-launcher (WHAT)

LTI-tool (for demo perposes)

Resources

- https://www.surf.nl/regie-op-de-digitale-leeromgeving/techniek (Dutch)
- https://drieam.github.io/LtiLauncher/
- https://site.imsglobal.org/certifications/drieam/lti-launcher
- https://www.imsglobal.org/spec/lti/v1p3/

Questions

Questions

- Would you be interested in further developing of the LTI launcher
- How is LTI implemented in Swedish HE with regards to data provisioning and GDPR

Ronald Ham

E-mail:
ronald.ham@surf.nl

www.surf.nl

Driving innovation together

Driving innovation together

SURF