

Server Structure

Language :
NodeJS with express

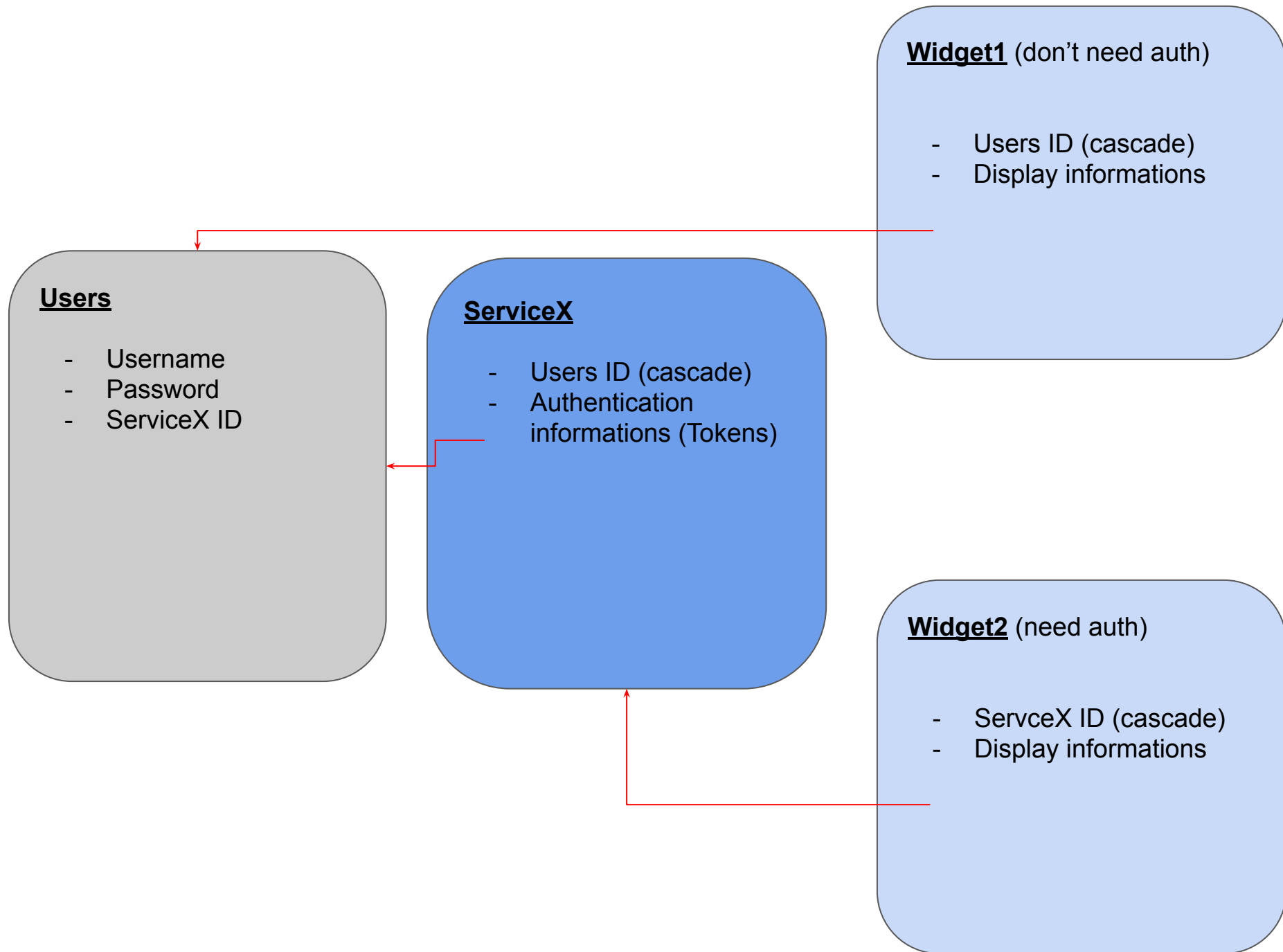
View engine:
EJS

Database:
PostgreSQL

Endpoint :

- GET / Display the login view
- GET about.json Return informations about services / widgets
- POST auth/ Authenticate the users with her user / pass
- GET auth/<service name> Redirect to the service connection page
- GET auth/<service name>/callback Endpoint used when creating service app
- POST data/<widget name> Return the widget data
- GET disconnect/ Logout the user
- GET disconnect/<service name> Revoke access to the specified service
- GET home/ Display the home view
- POST home/save Save the instances of the widgets from the home view into the db
- GET profile/ Display the profile view
- GET register/ Display the register view
- POST register/ Create new user

Database structure



Users

- Username
- Password
- Coinbase ID
- Twitter ID

Coinbase

- Users ID
- Access Token
- Refresh Token

Exchange (don't need auth)

- Users ID
- Currency1 (String)
- Currency2 (String)

Wallet (need auth)

- Coinbase ID
- Currency (String)

Twitter

- Users ID
- Token
- Token Secret

Timeline (need auth)

- Twitter ID
- Type (String)

How to add a new service

Create your own **app** on the desired Service in order to get an authentication page

Add the required **endpoint** to the server
(cf: Server Structure)

Add the Service into the **database** (cf : Database Structure)

When `**auth/<service name>/callback**` get hit,
insert new row to the service database

Some **modification** to the following files are
required in order to display the new Services :
home.js / home.ejs
profile.js / profile.ejs

How to add a new widget

Add the service to the **database** (cf : Database Structure)

Add the required **endpoint** to the server
(cf: Server Structure)

Modify home.ejs / home.js in order to view the new Widgets

Create the following **function** :

<i>create_<widget name>_widget</i>	Inject html with unique id code and appropriate class name to the board div
<i>get_<widget name>_widget</i>	Return the element needed for the database (timer, data...) from the class name
<i>save_<widget name>_widget</i>	Call the timer addTimer func with the id+ time + data + a callback function
<i>update_<widget name>_widget</i>	this is the callback function, will be called automatically, need to retrieve data from data/<widget name> endpoint and inject it to the widget