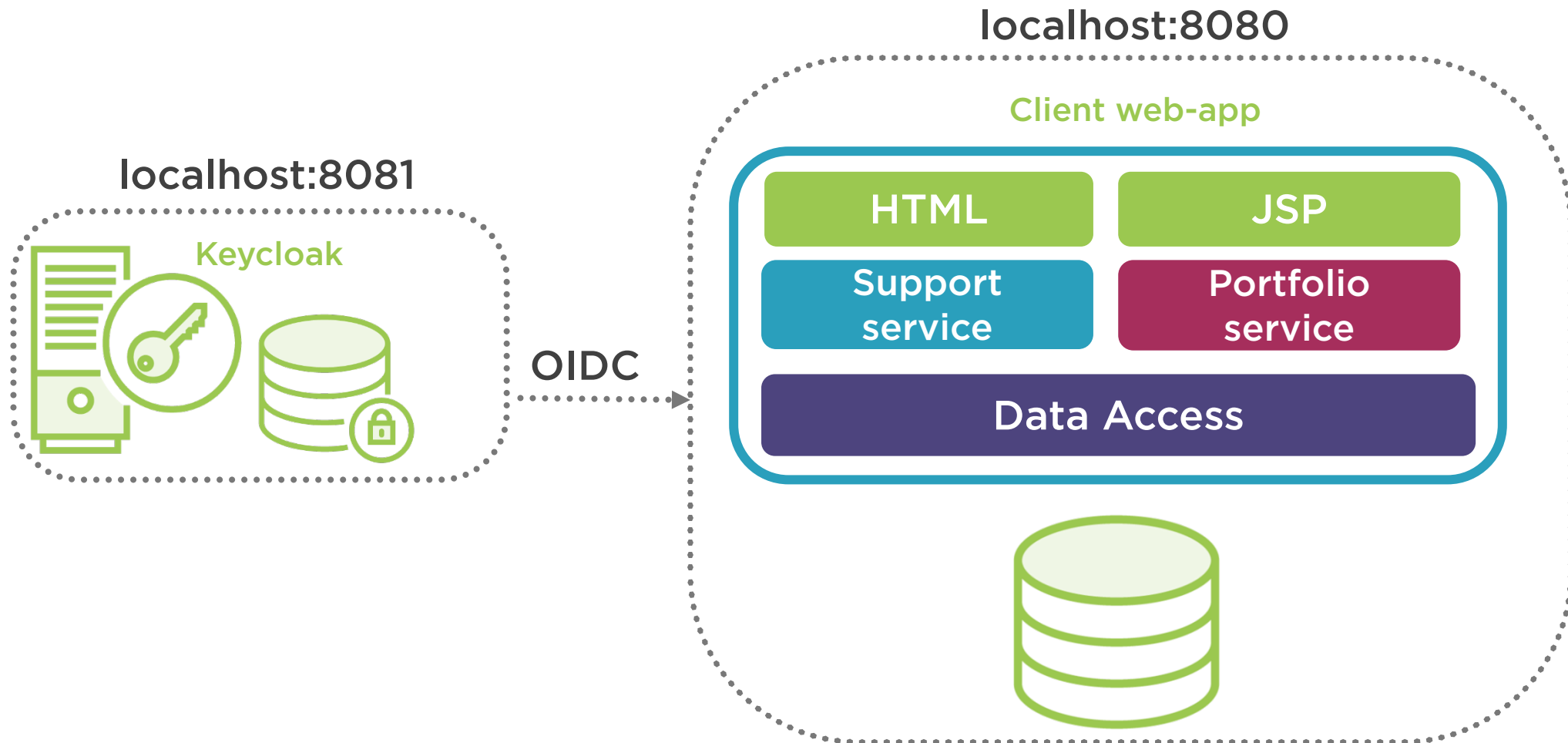
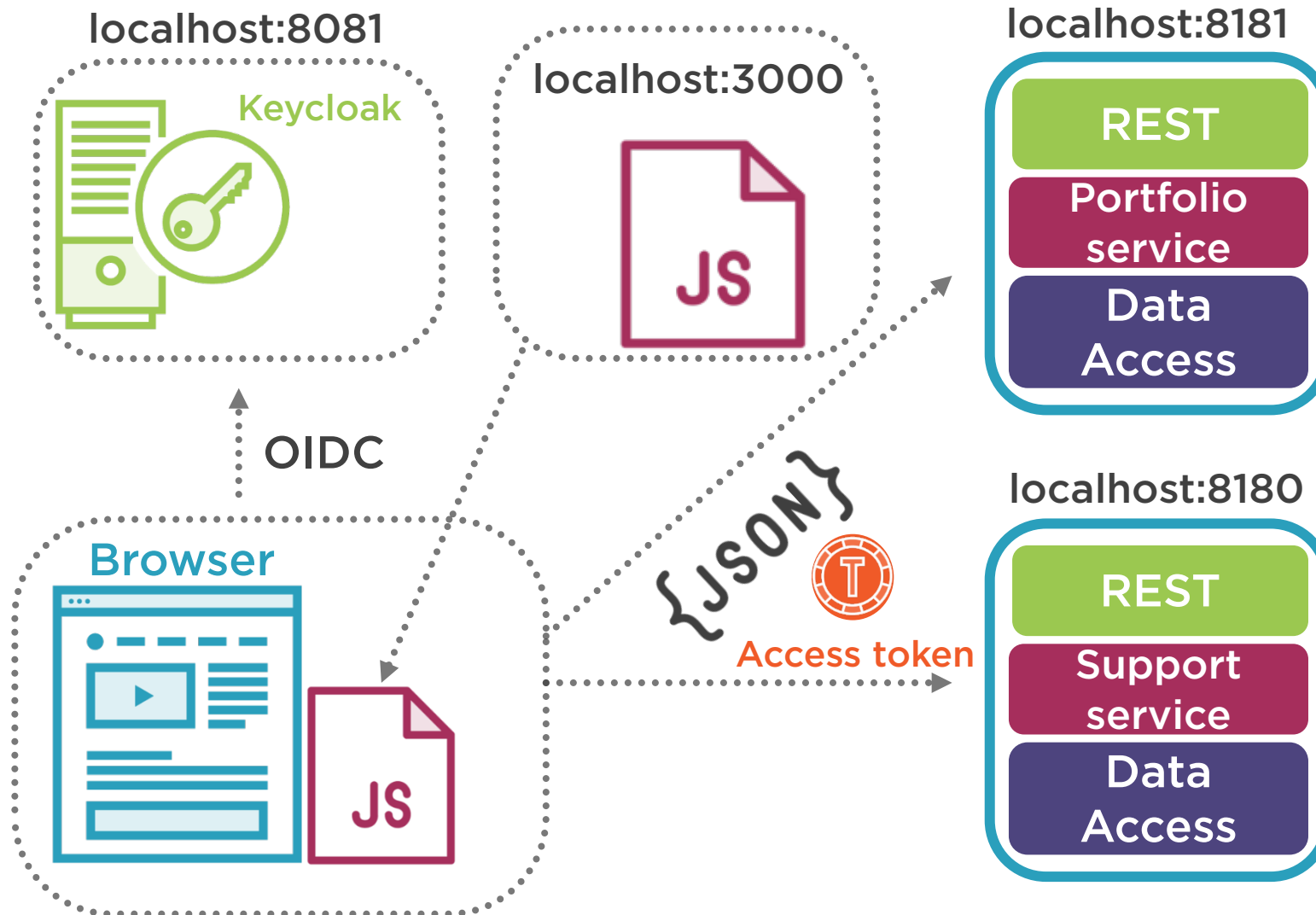


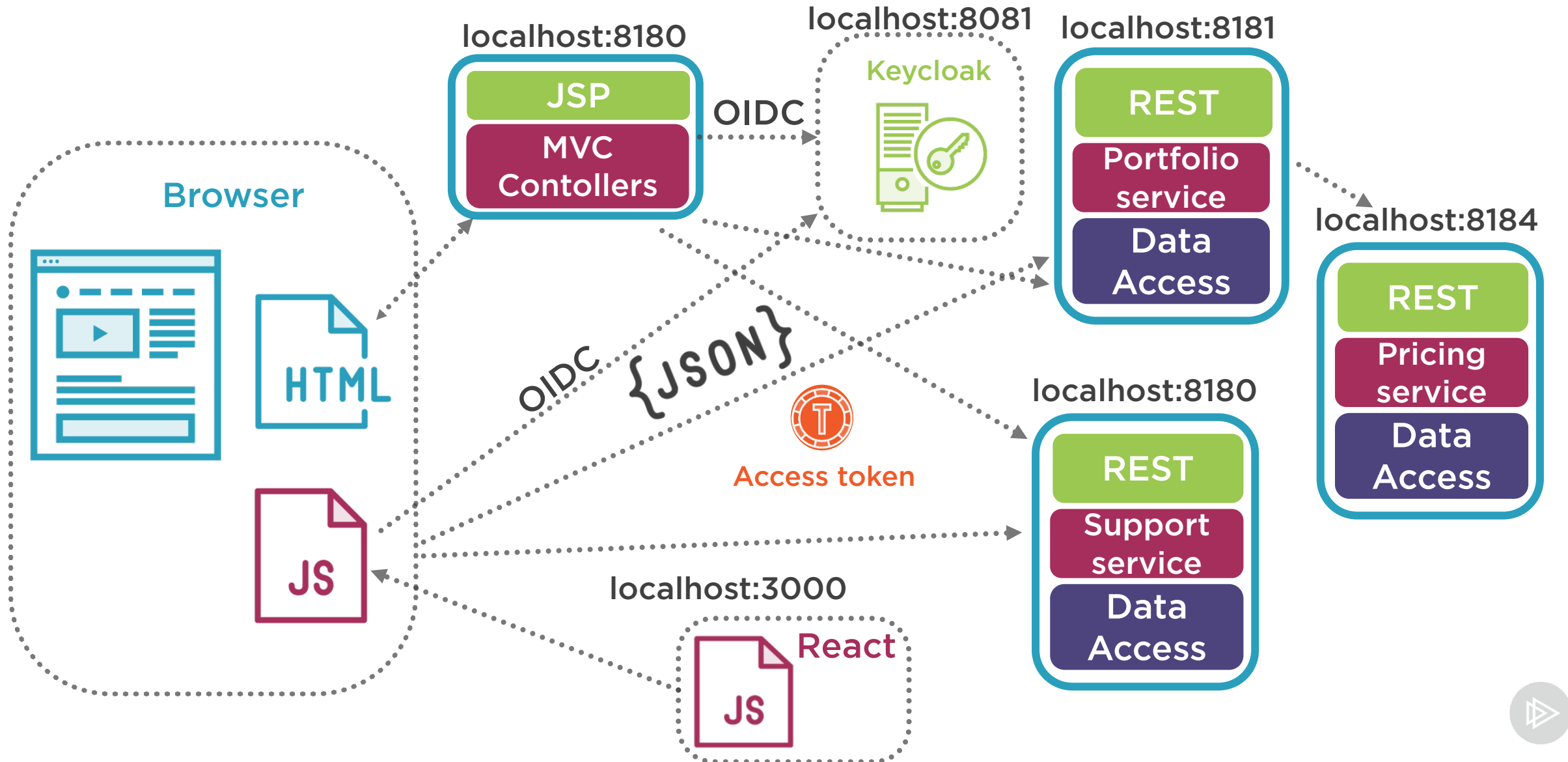
MVC Client Architecture



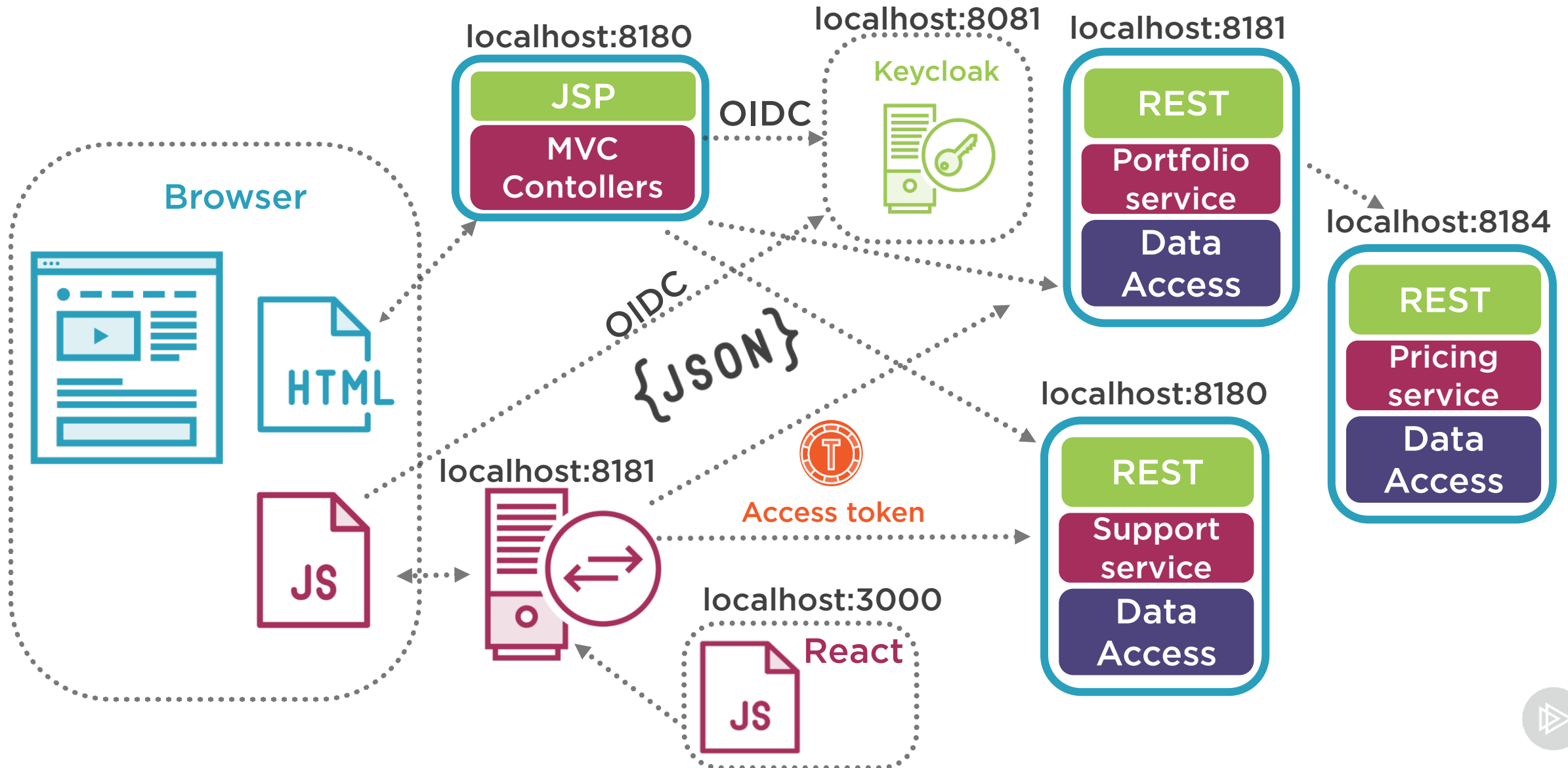
React Client Architecture



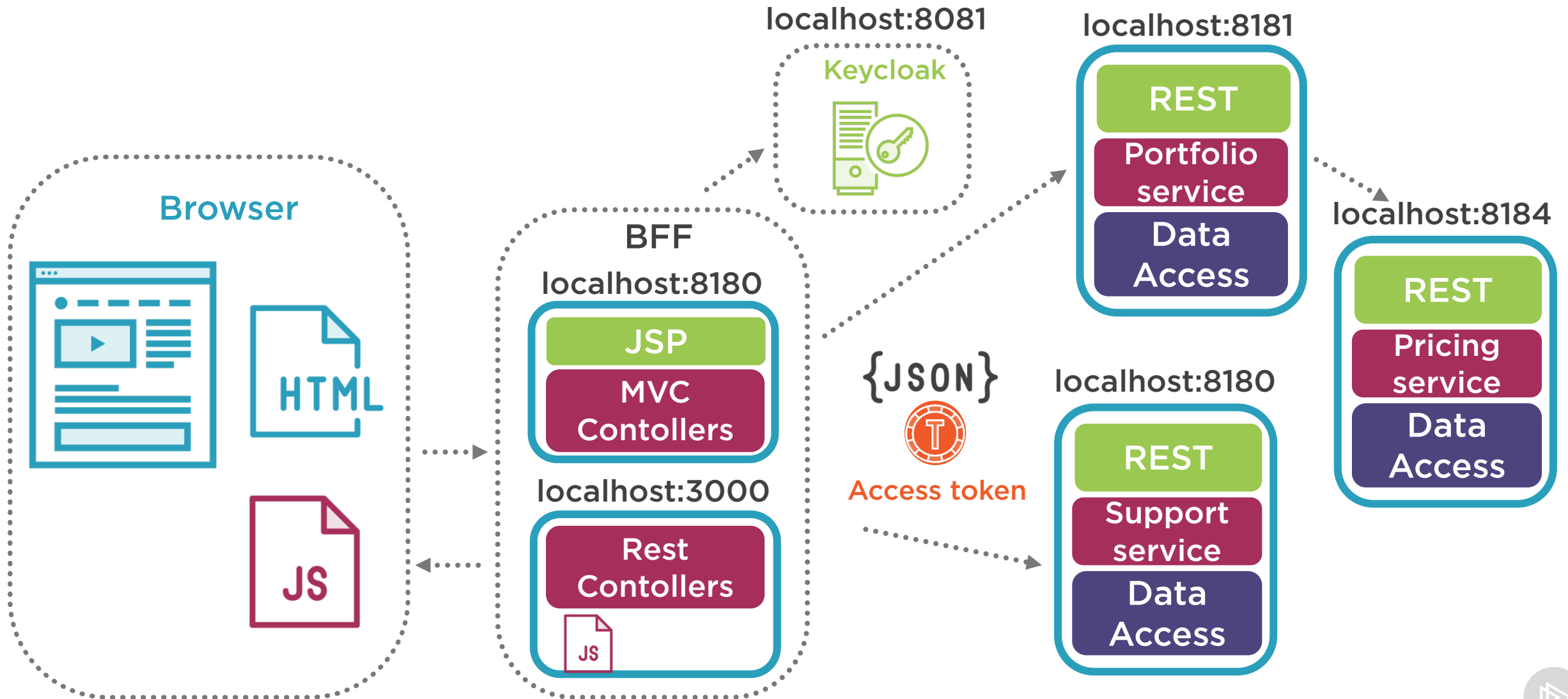
Microservices Architecture



Backends for Frontends pattern (BFF)






Backends for Frontends pattern (BFF)



Token Relay








Expiry time
Name: Victoria
Email: vic@email.com
Address: 1 Smith St, Lnd
Scopes: portfolio
support
pricing
Roles: portfolioUser
supportAdmin

Identity Provider



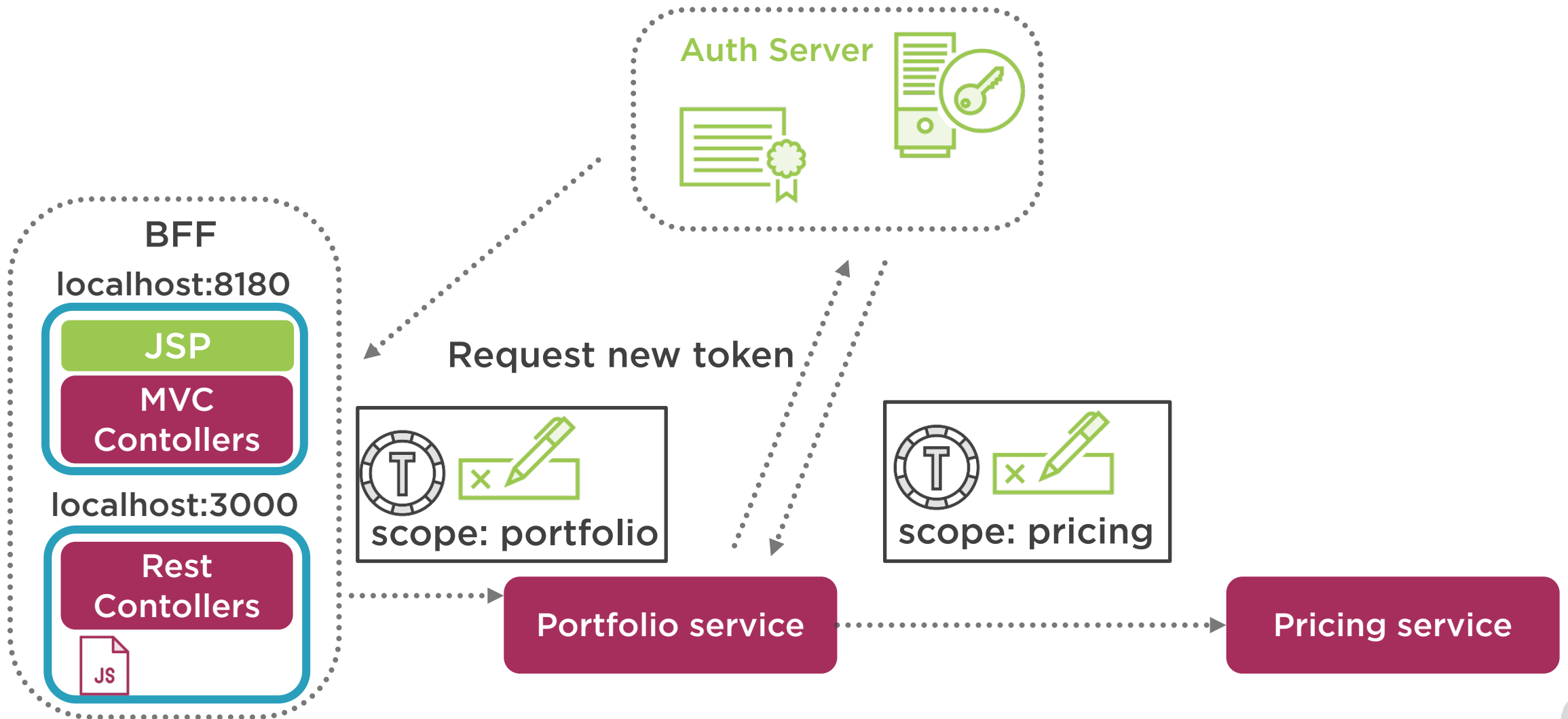
Portfolio service

Support service

Pricing service

LOGS

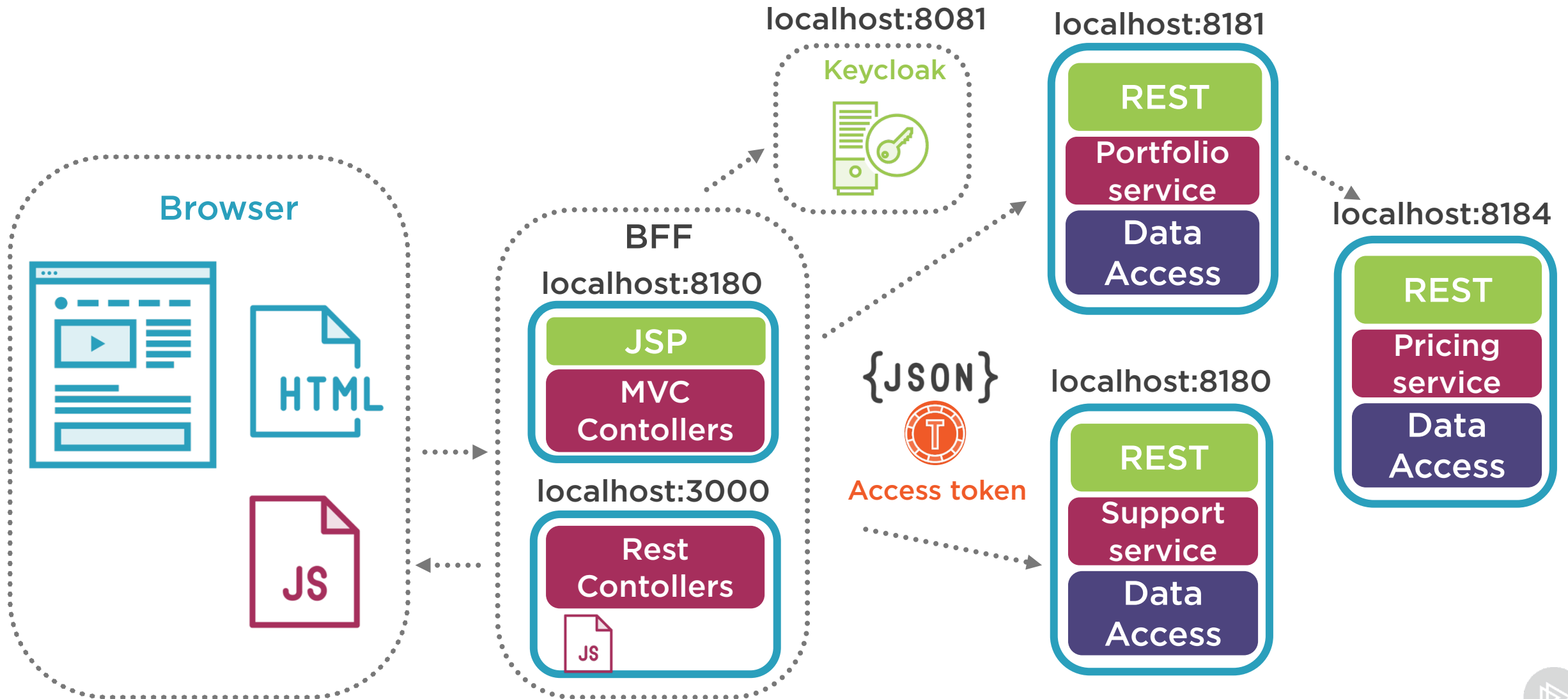




Client Credentials Grant



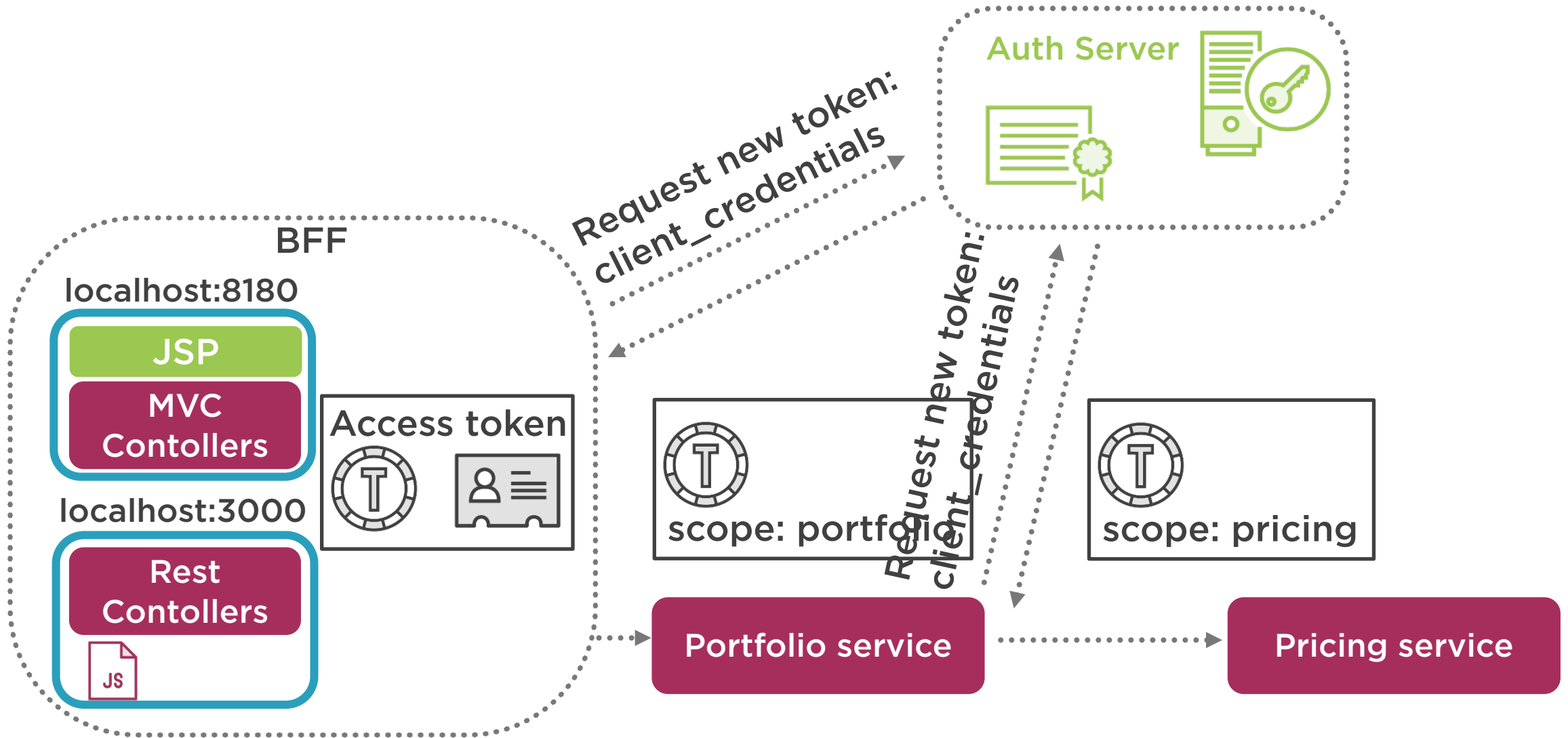
Client Credentials Grant



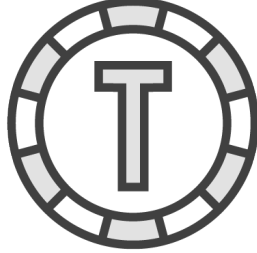
Client Credentials Grant



Client Credentials Grant



Access token



Expiry time
username: victoria
Email: vic@email.com
Address: 1 Smith St, Lnd
Scopes: portfolio
support
pricing
Roles: portfolioUser
supportAdmin

BFF Client

Portfolio
web-app



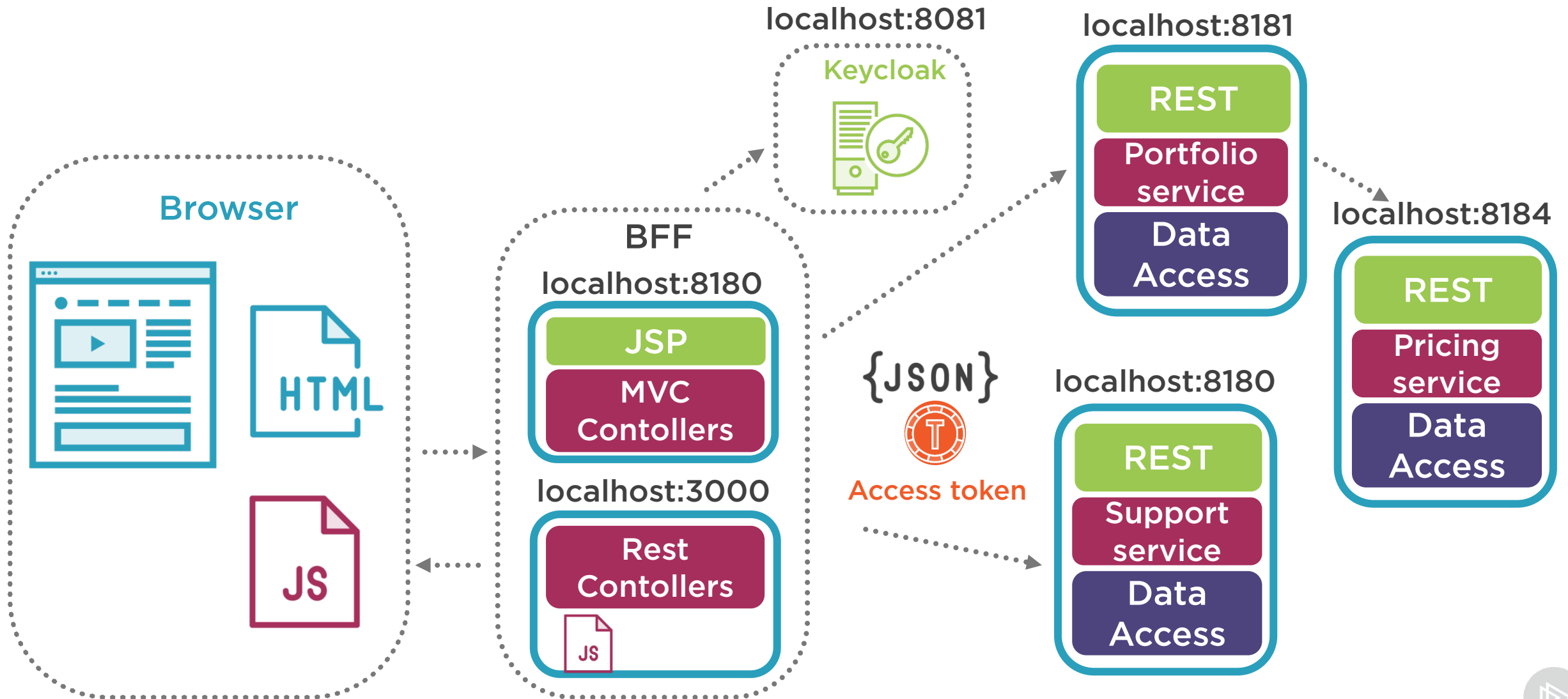
Auth Server



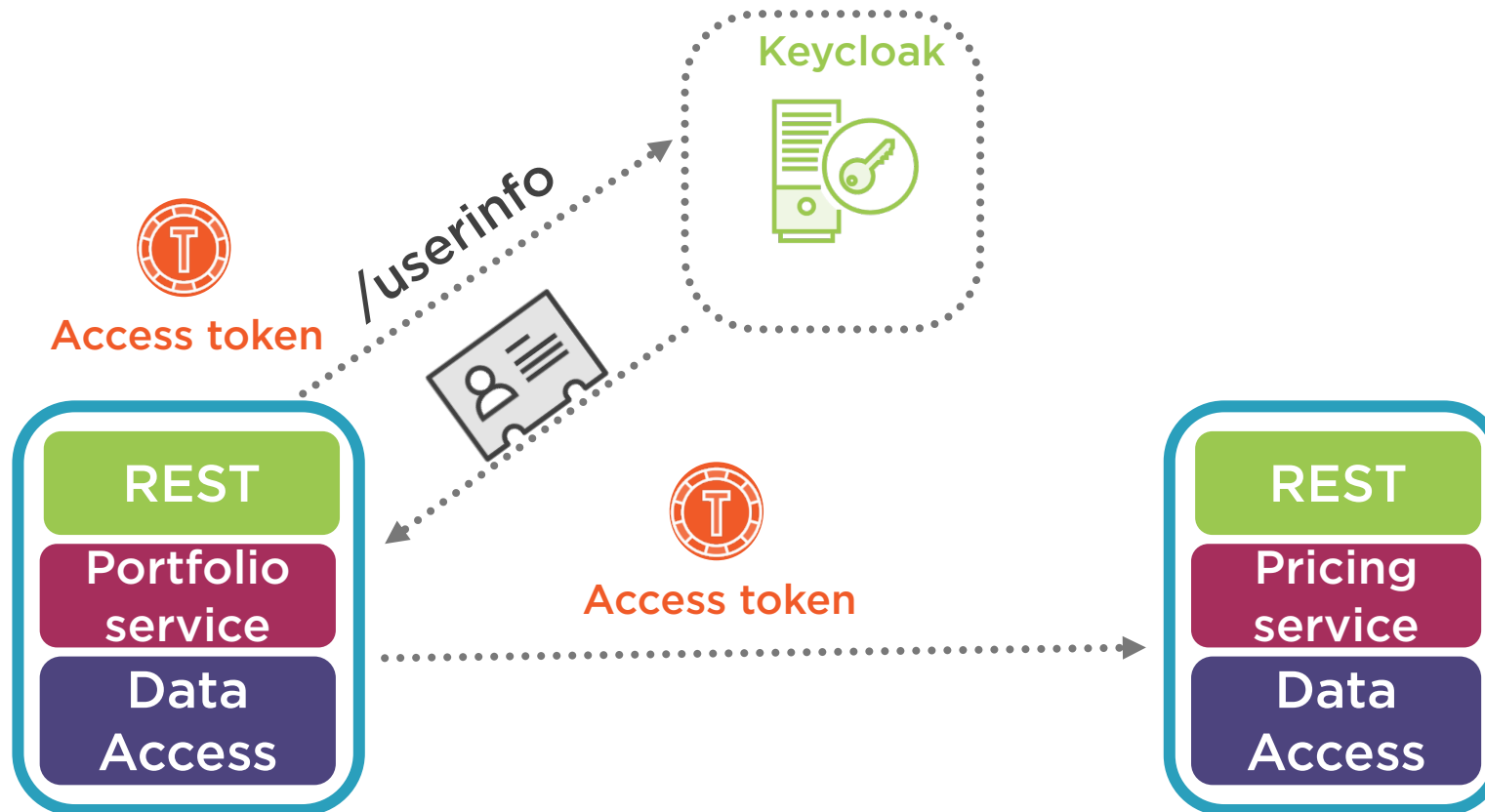
WebClient vs. RestTemplate



WebClient with Token Relay



By Reference User Claims



Client Credentials Grant via RestTemplate



Module Complete



What you learned:

- Implement Client-credentials grant
- Microservice security patterns

Some key take-aways:

- Keep sensitive content in the token to a bear minimum
- Most systems won't need the scalability
- Consider encryption of token if it contains sensitive data
- Access tokens can be used against the userinfo endpoint by the bearer

