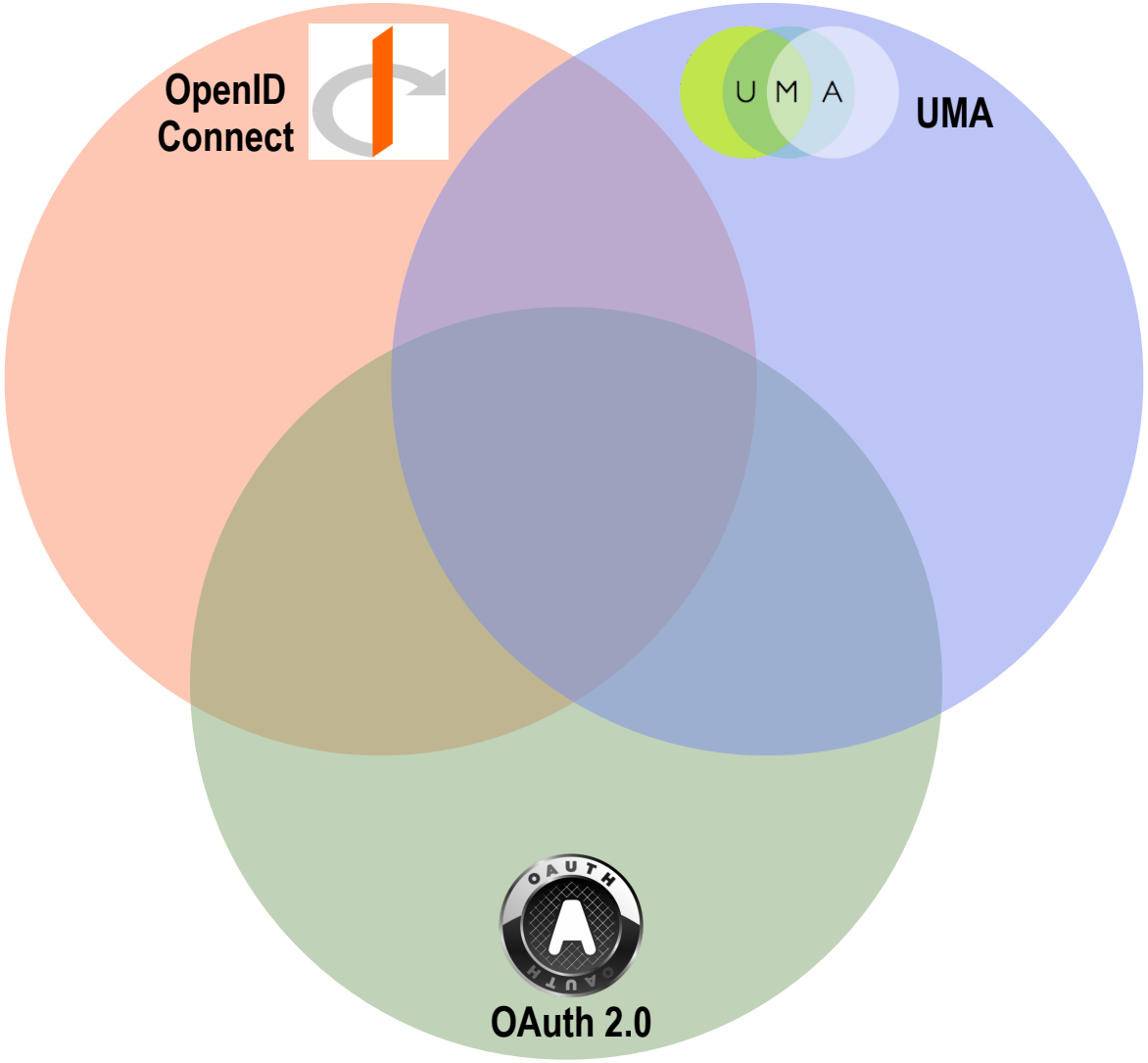
Access Control Venn Infographics

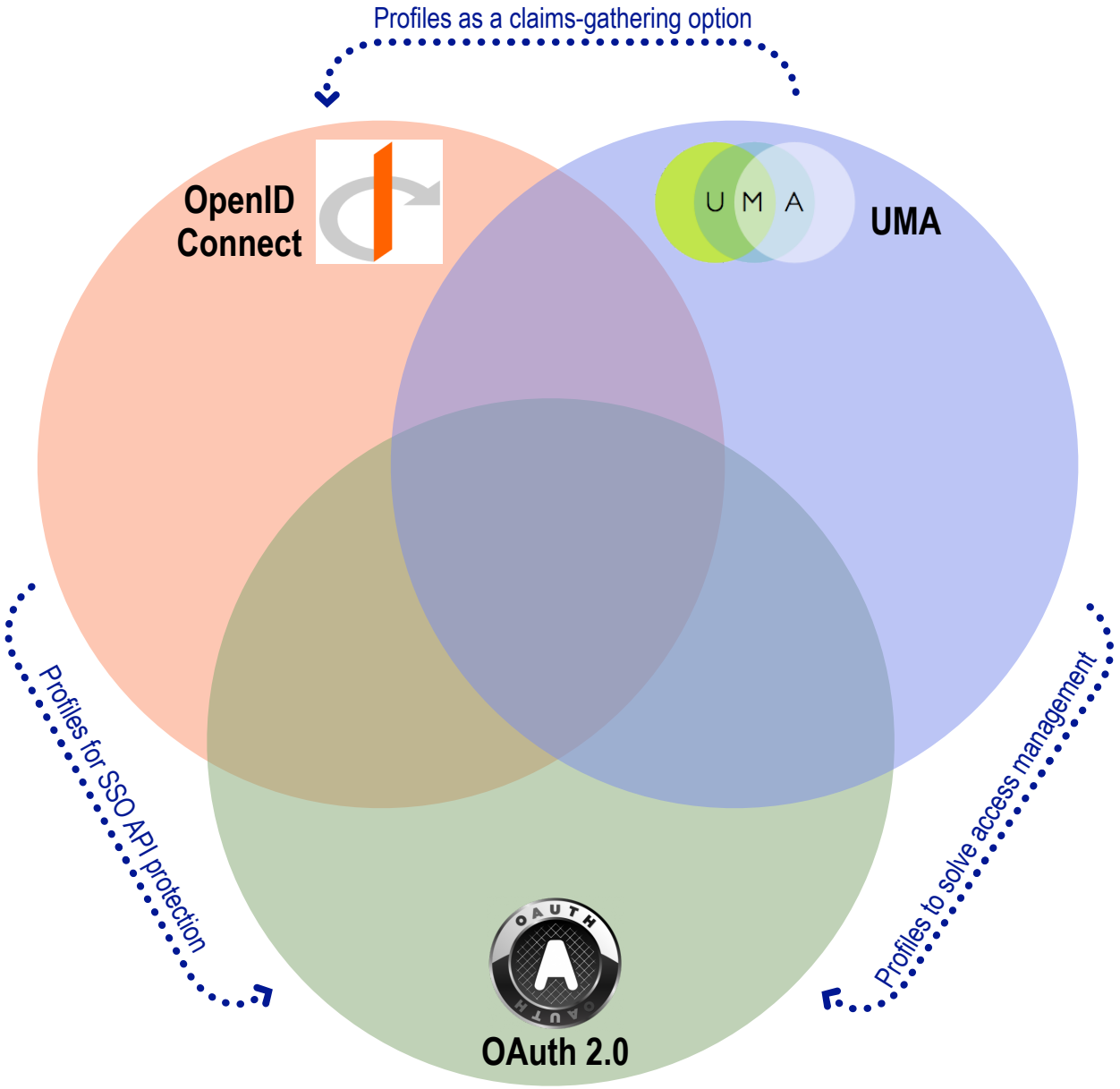
UMA Work Group

16 July 2013

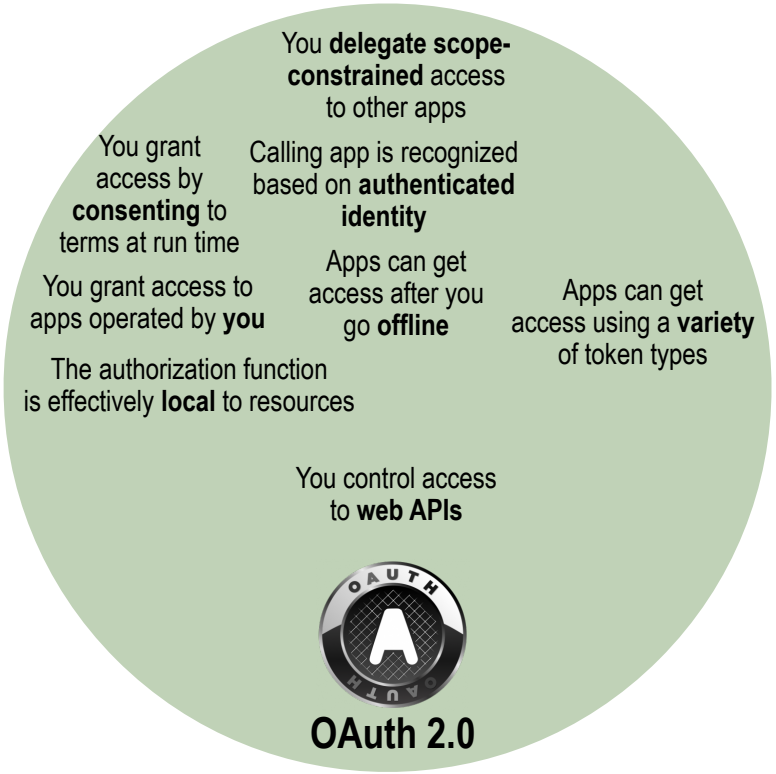
Comparing three technologies



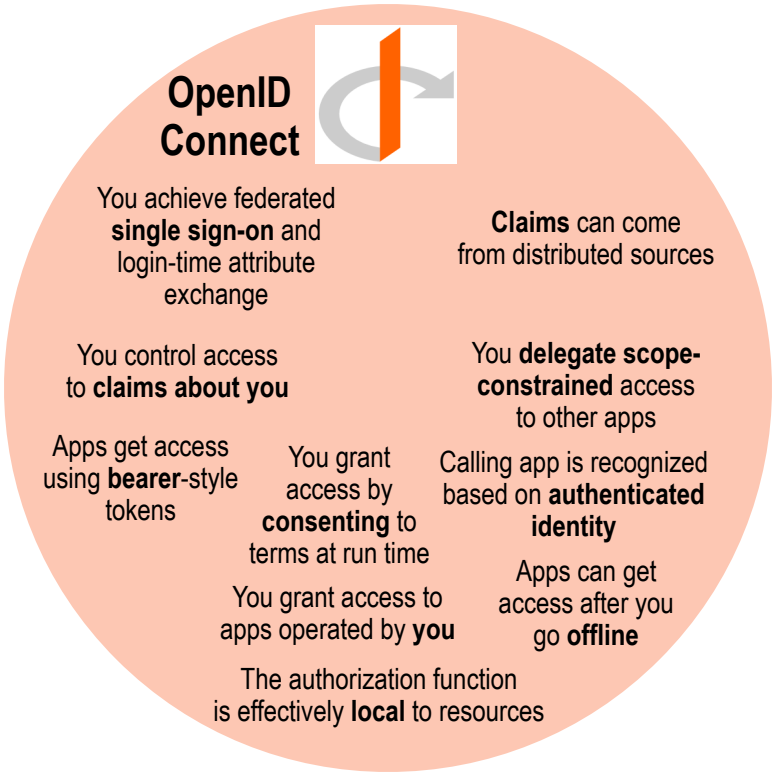
Their relationships



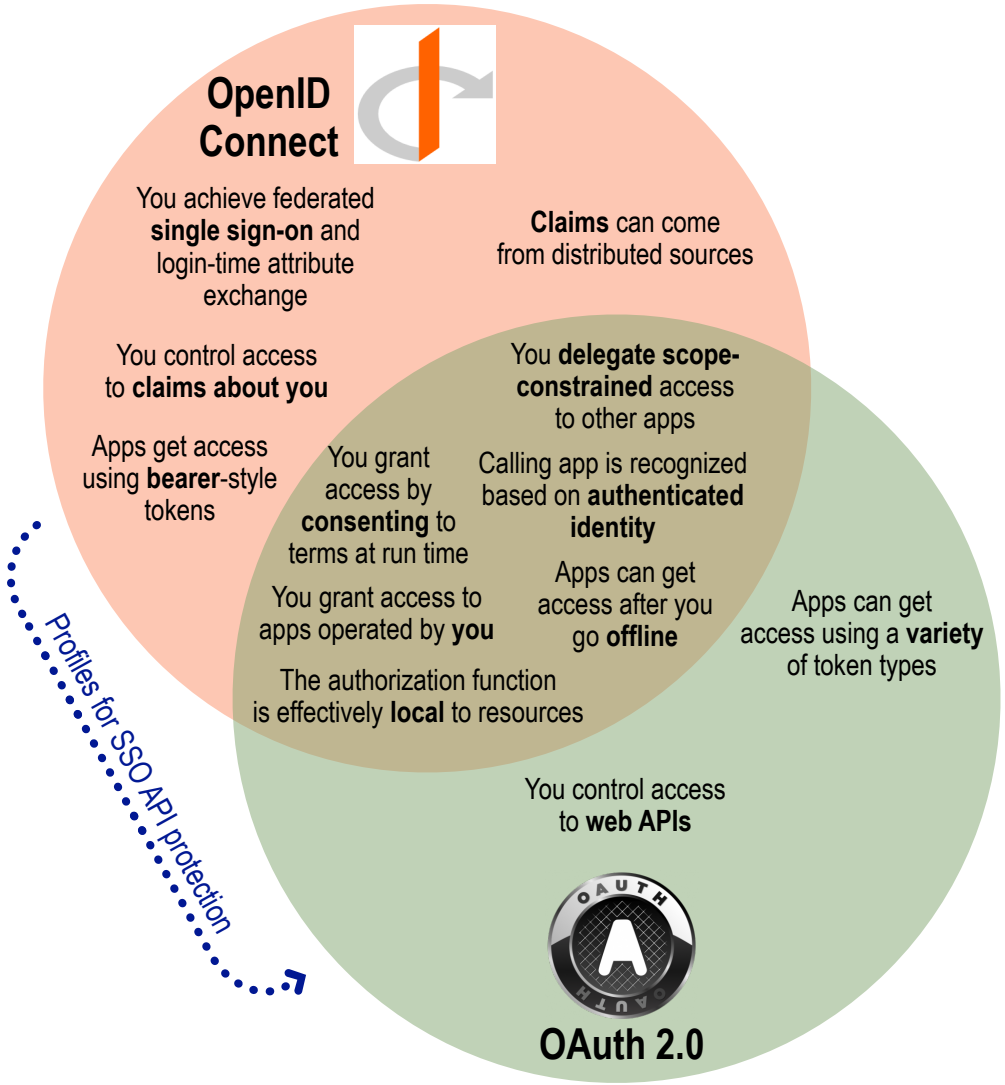
OAuth in a nutshell (as usually deployed)



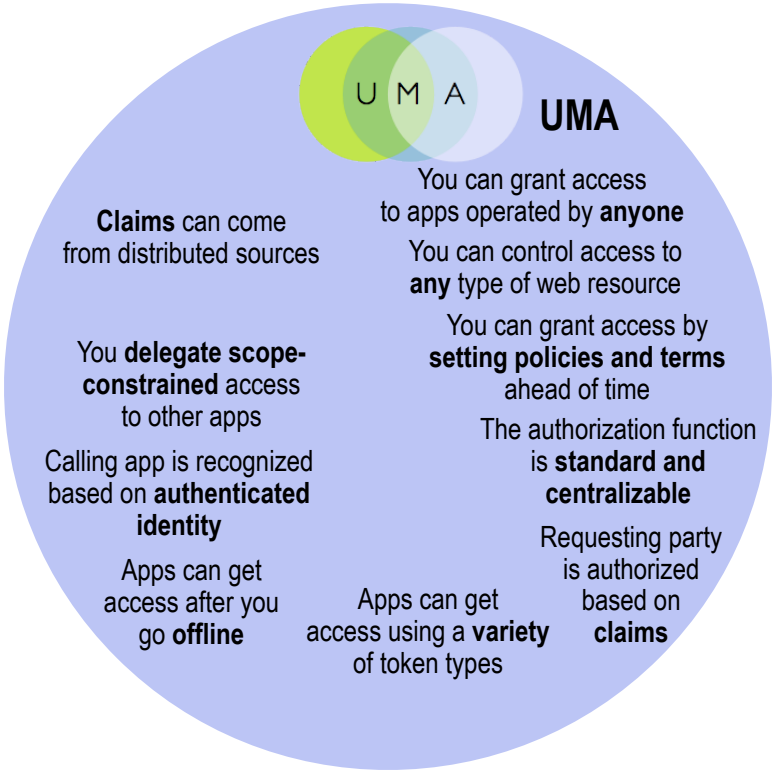
OpenID Connect in a nutshell



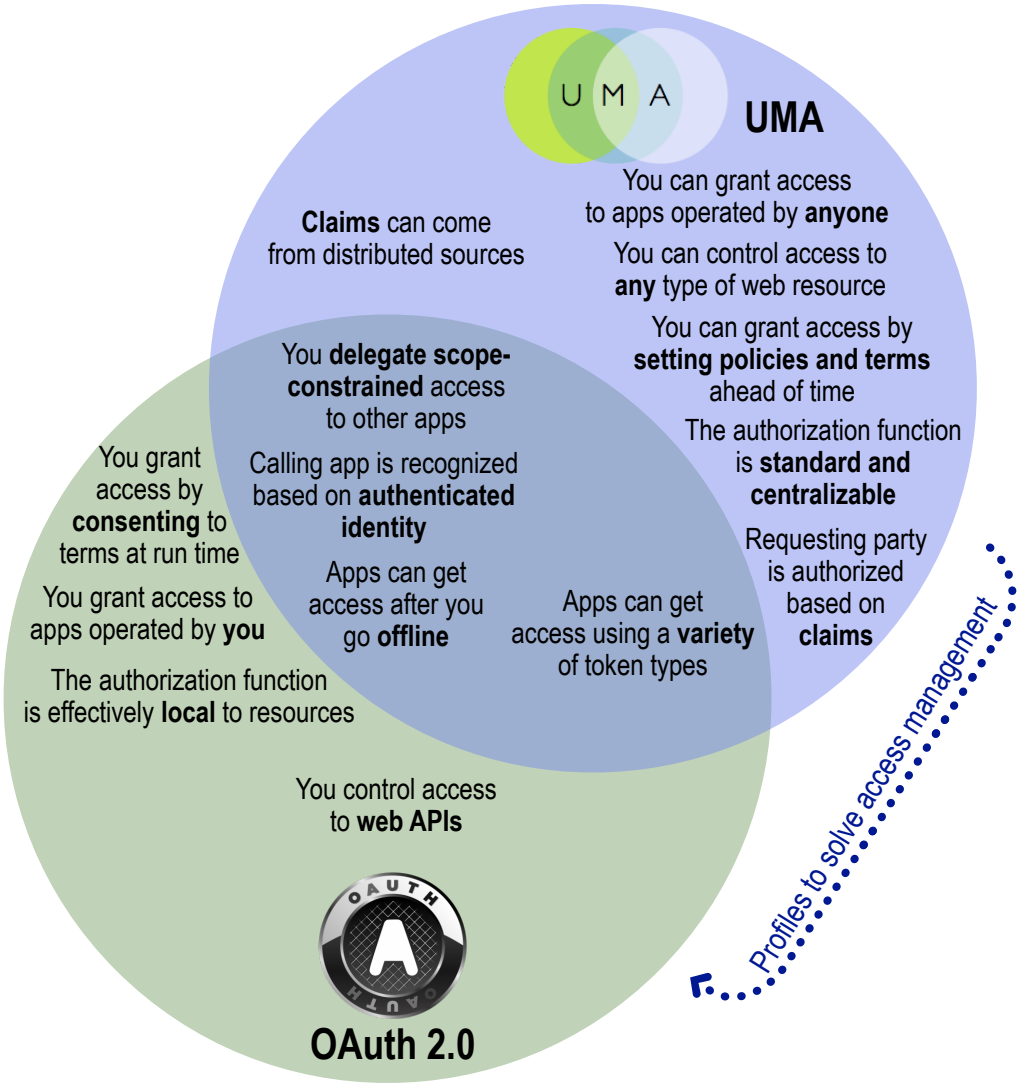
What OAuth and OpenID Connect share



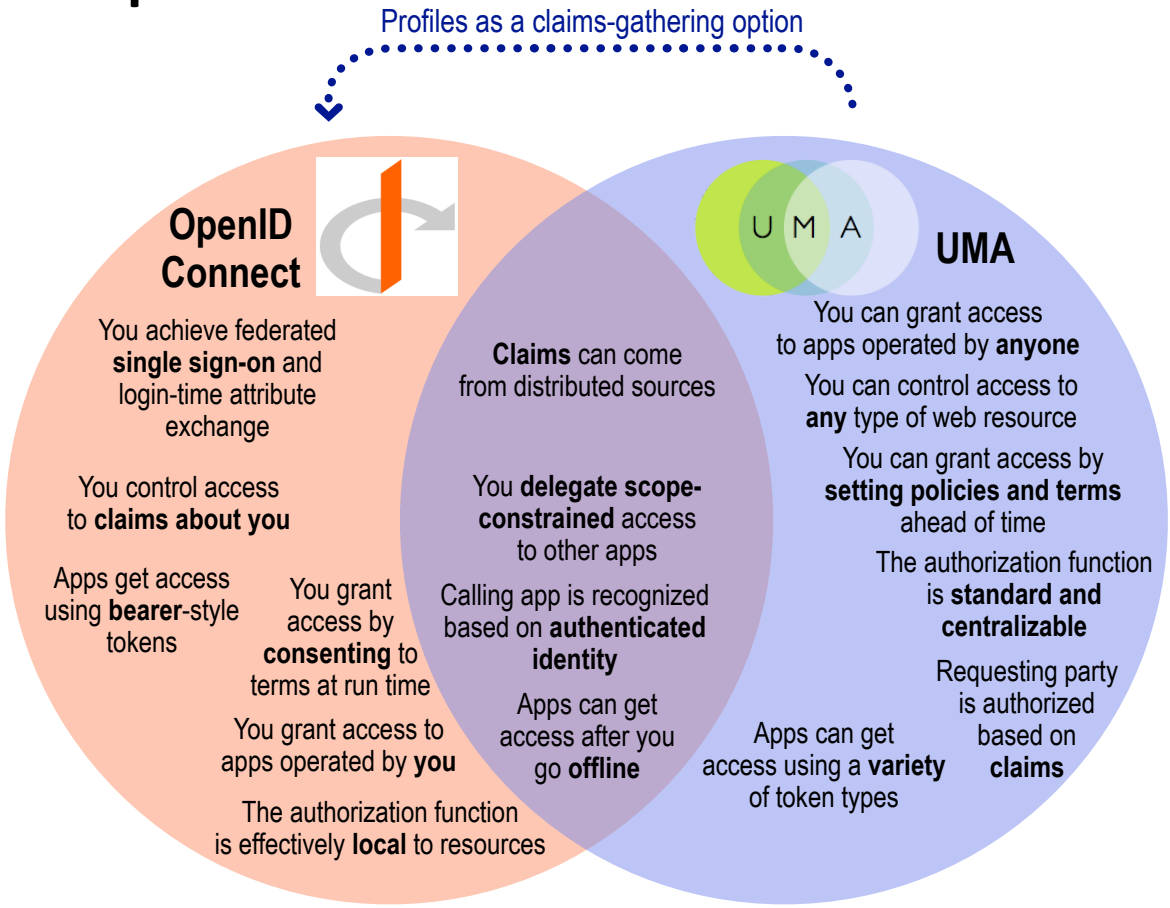
UMA in a nutshell



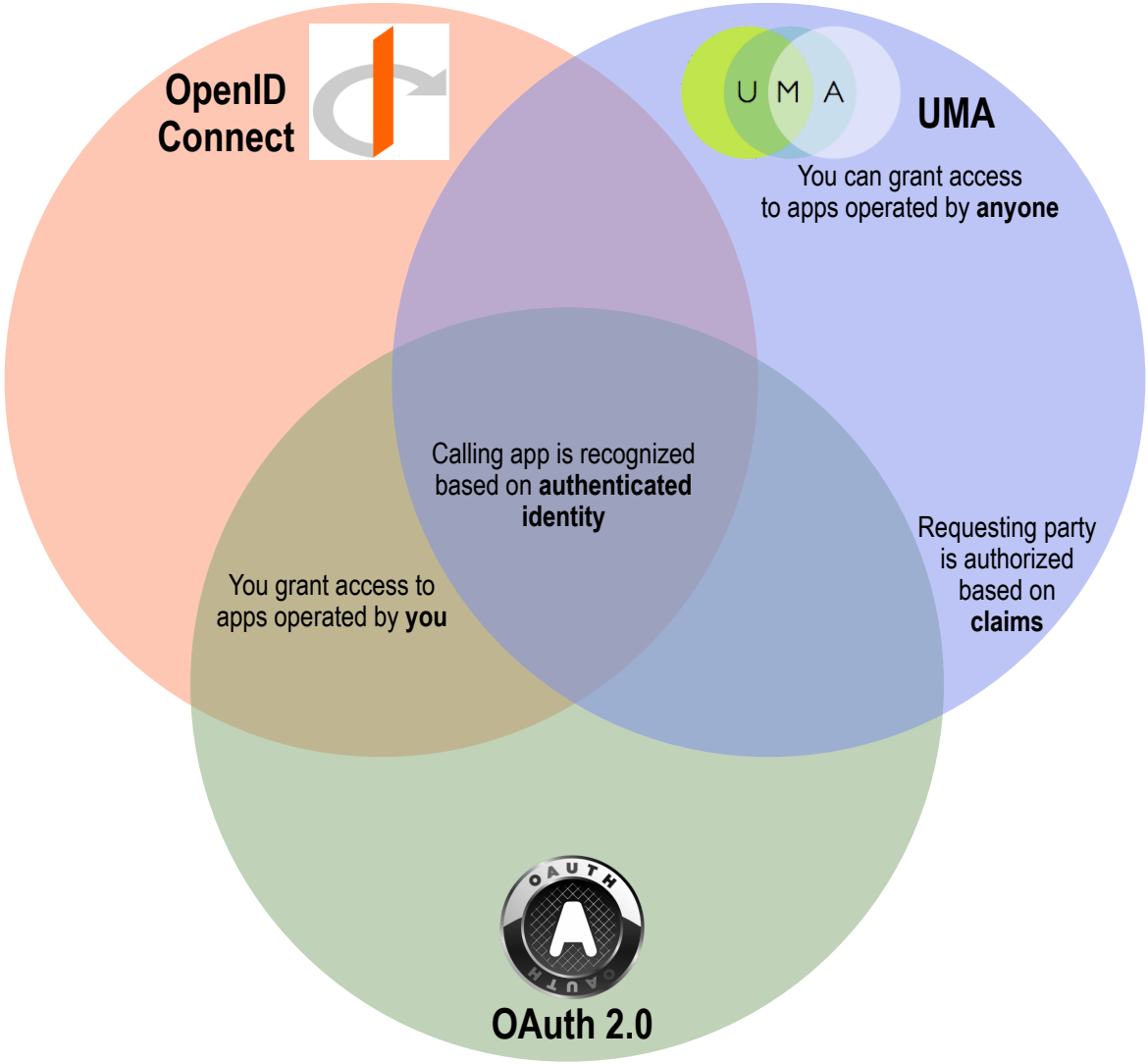
What OAuth and UMA share



What OpenID Connect and UMA share

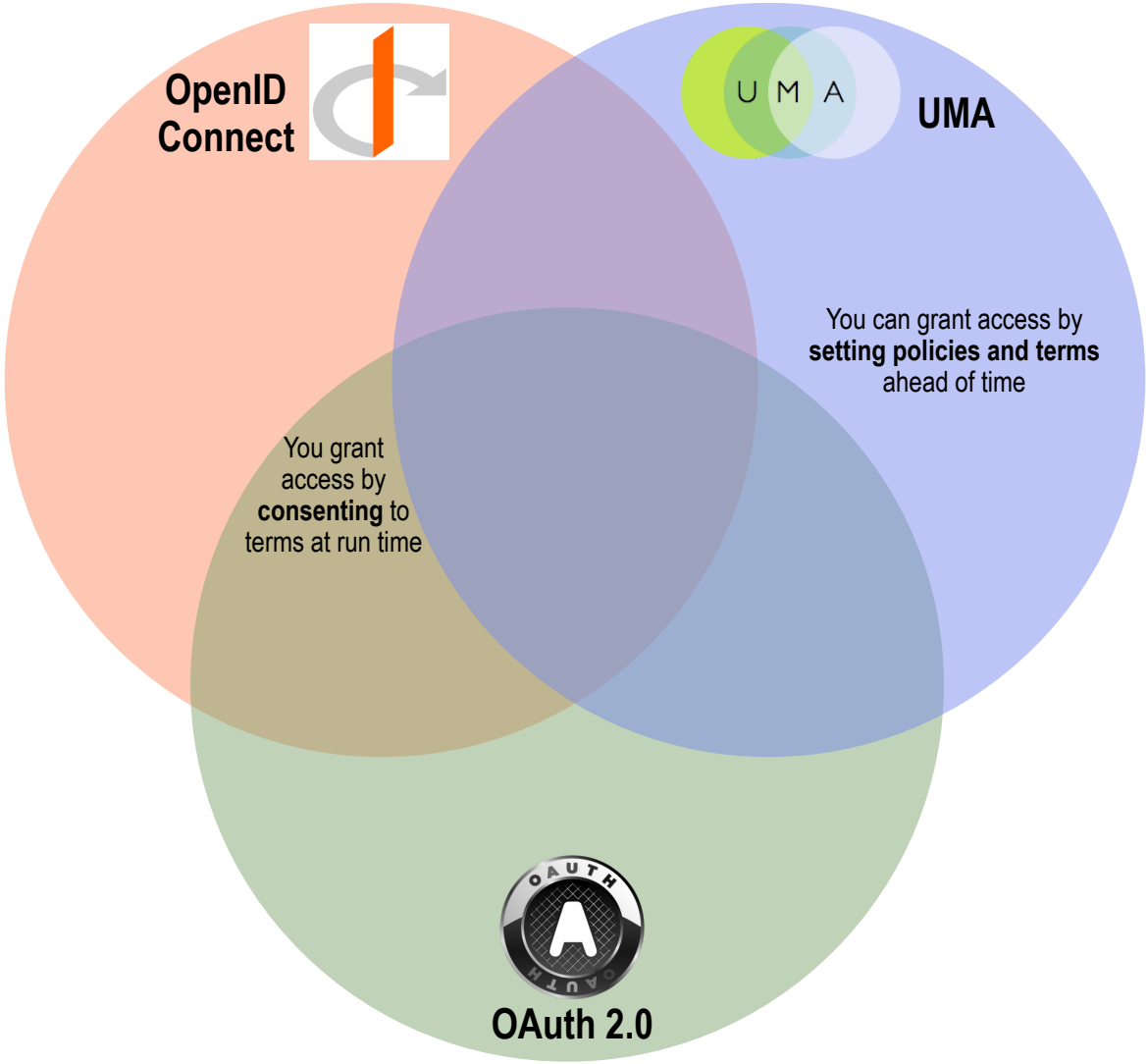


Controlling access: by what/whom?

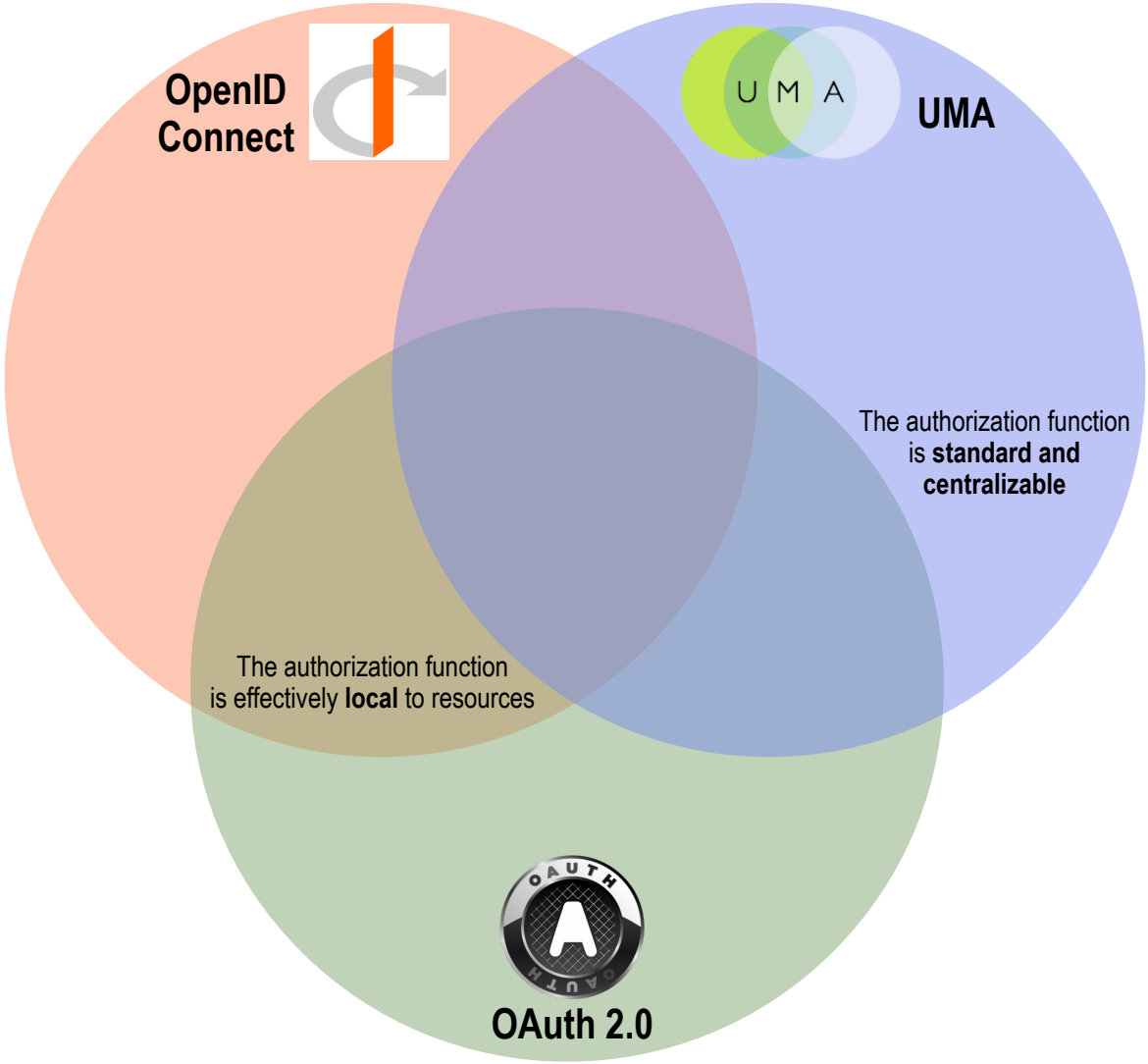


Controlling access: to what?

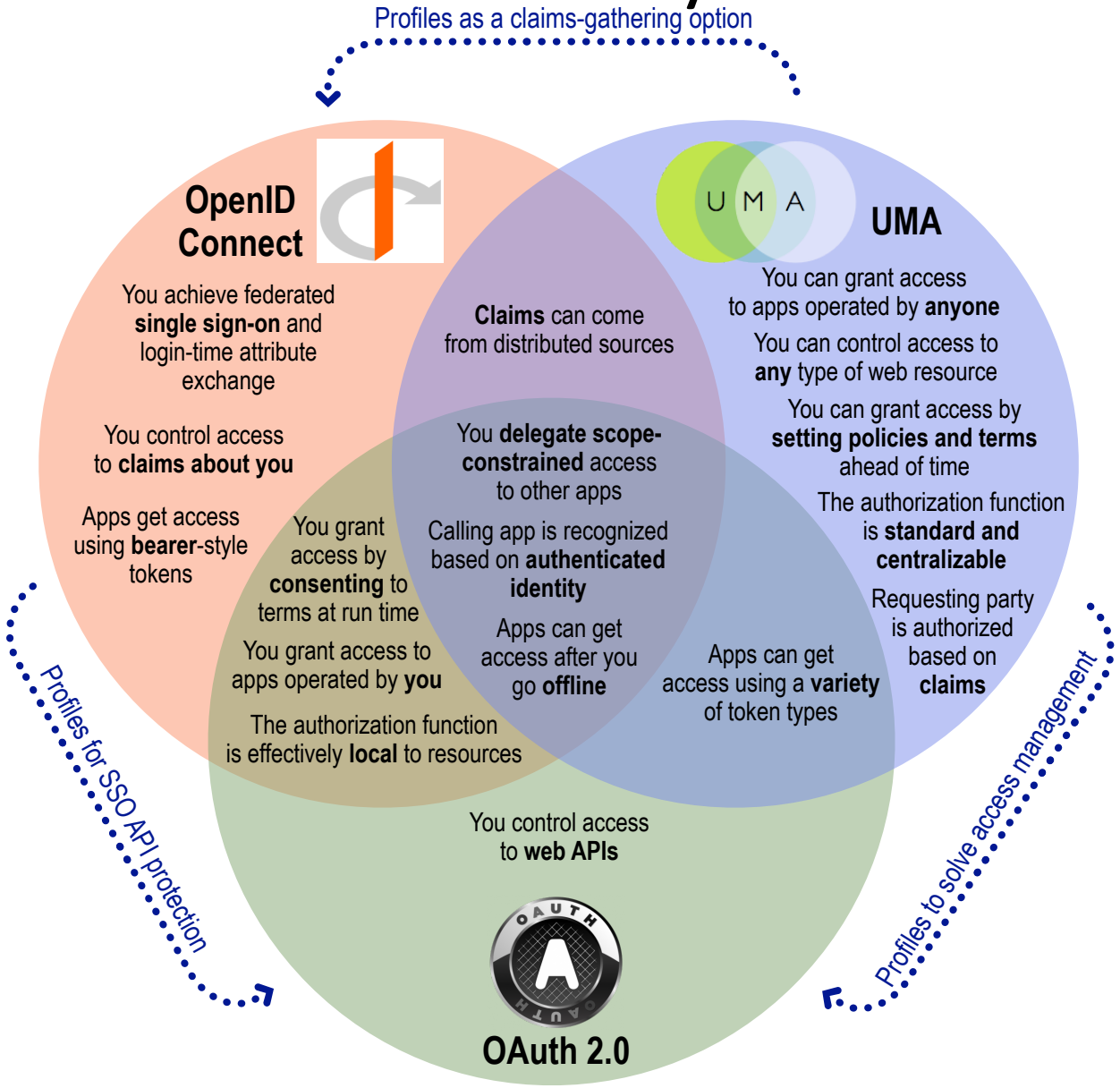
Controlling access: by what means?



Authorization function: how is it coupled?



Summary


Profiles as a claims-gathering option
OpenID Connect
You achieve federated **single sign-on** and login-time attribute exchange
You control access to **claims about you**
Apps get access using **bearer**-style tokens
UMA
You can grant access to apps operated by **anyone**
You can control access to **any** type of web resource
You can grant access by **setting policies and terms** ahead of time
The authorization function is **standard and centralizable**
Requesting party is authorized based on **claims**
Claims can come from distributed sources
You **delegate scope-constrained** access to other apps
Calling app is recognized based on **authenticated identity**
Apps can get access after you go **offline**
You grant access by **consenting** to terms at run time
You grant access to apps operated by **you**
The authorization function is effectively **local** to resources
Apps can get access using a **variety** of token types
You control access to **web APIs**
OAuth 2.0
Profiles for SSO API protection
Profiles to solve access management