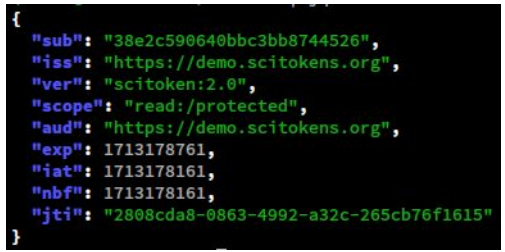
Grand Unified Token (GUT) profile

Mischa Sallé, Nikhef

HEPiX, Paris, 17 April 2024

Background 1/2

- Move towards token based authentication/authorization: motivated by e.g. social providers: OAuth2, OpenID-Connect
- Replacing
  - X.509/VOMS-proxies (RFC3820) in grid/HTC
  - SAML2 for federations/NRENs
- Several groups started using JSON Web Tokens as access tokens (this is before RFC9068)



- Typically requires profiling what is put in the tokens: which claims, values, etc.

Background 2/2

Several parties created JWT-based OAuth2 profiles:

- SciTokens: one of the 1st to create profile for “our” distributed infrastructures
  - using “capabilities”: not interested in *who*, but *what* is allowed

- WLCG: profile based on experience with VOMS proxies and SciTokens
  - support for both capabilities and users/groups
  - aims to be interoperable profile for WLCG VOs.
  - motivated by the EoL of Globus

- AARC projects: create guidelines / profiles to be interoperable between infrastructures
  - support for both capabilities and users/groups
  - strong focus on interoperability

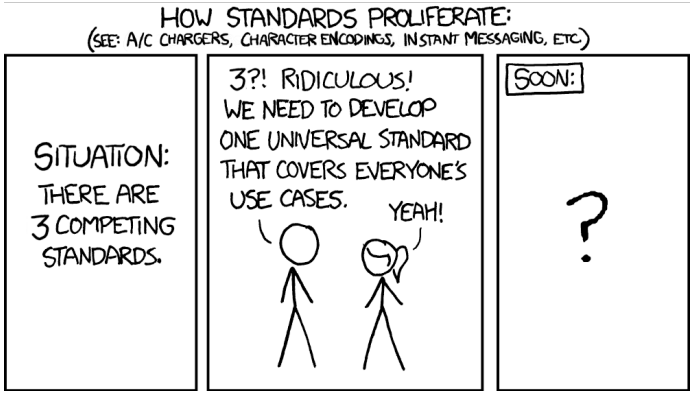
Introducing GUT 1/2

- 3 profiles is bad for everyone:
  - Developers
  - System administrators
  - Security

➔ Bold plan: try to unify these 3 profiles into a

***G**rand **U**nified **T**oken (GUT) profile*

(Risk: we'll have 4 profiles[1])



[1]adapted from https://xkcd.com/927/

Introducing GUT 2/2

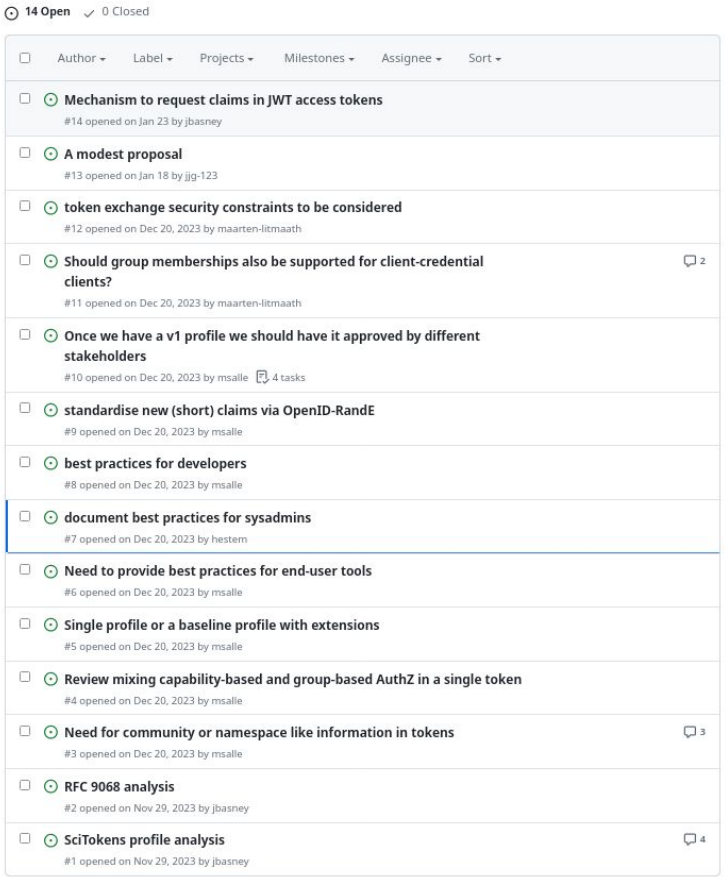
- Crucial to have unanimous support from all three groups
- Formed a completely neutral group of interested people from:
  architects, developers, sysadmins, (expert) users, members of the different profile groups
- Each group should decide how to implement/migrate to the new profile
- Have access to standardisation bodies, such as OpenID RandE working group for standardising claim names etc.

GUT Goals

- Form a single basic profile
- Possible to have optional add ons
- Ideally simplify existing profiles (complications ⇨errors)
- Look for related groups we might have missed
- Standardise as much as we can in official channels/working groups
- Get approval from all the 3 profile groups
- Get approval from all the main software tools/stacks for implementing:
  - AAI solutions: Indigo-IAM, EGI-Checkin, CILogon, EduTeams, …
  - HTC CEs: HTCondor, ARC-CE,...
  - Storage solutions: dCache etc.

Current status

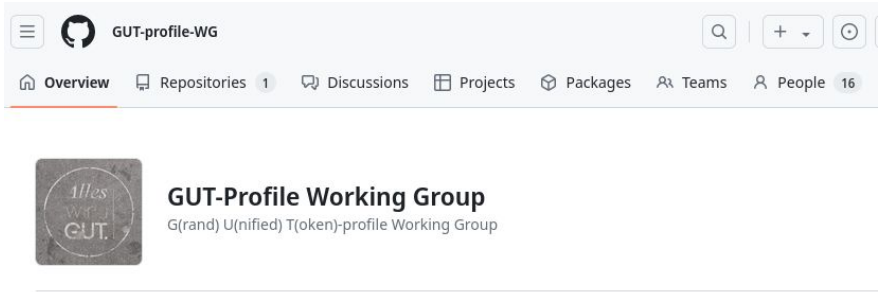
- Several meetings so far
- Presented some comparison talks
- Identified a number of issues ("points of harmonisation")
- Started working on first issue: the need for a claim to identify the VO, community etc.
- Difficulties ahead, e.g.
  - Parametric scopes (too complicated)
  - Authorization too fine grained



How we work

- Open to everyone, aim for a diverse global mix of experts
  Architects, deployment experts, developers, sysadmins, security experts

- Github organisation and repository
  - https://github.com/GUT-profile-WG/GUT-profile
  - Using issues: topic and discussion
  - Once in a more crystalised form also PR



- Google docs:
  - Running collaborative meeting notes
  - Quick drafting with suggestions

- Mailing list:
  - https://mailman.nikhef.nl/mailman3/postorius/lists/gut-profile.nikhef.nl/

- Roughly monthly meetings:
  - https://indico.nikhef.nl/category/93/

GUT-Profile - links and related documents

Some relevant doc links:

- GUT Profile:
  - https://github.com/GUT-profile-WG/GUT-profile
  - https://mailman.nikhef.nl/mailman3/postorius/lists/gut-profile.nikhef.nl/
  - https://indico.nikhef.nl/category/93/
- SciTokens:
  - https://scitokens.org/technical_docs/Claims
  - Background and further reading, see also https://scitokens.org/ and https://sciauth.org/
- WLCG profile:
  - Develop version: profile.md at https://github.com/WLCG-AuthZ-WG/common-jwt-profile
  - V1: https://zenodo.org/records/3460258
- AARC profile:
  - AARC-TREE (no typo but still 3rd AARC project): https://aarc-project.eu/
  - AARC guidelines for groups: https://aarc-community.org/guidelines/aarc-g069/
  - AARC guidelines for capabilities: https://aarc-community.org/guidelines/aarc-g027/
  - …

Some technical points 1/2

- Parametric scopes:

  For example: `scope=storage.read:/dune storage.create:/dune/data`
  - alternative for lack of “claims request” in client libraries
  - puts a whole language in claim value
  - used by all 3 but not in the same way

- Standardising short claim names

  IANA registry (https://www.iana.org/assignments/jwt/jwt.xml) and OIDC R&E working group (https://openid.net/wg/rande/)

  For example:
  - **`vo`** claim (or whichever we name we decide to use)
  - **`ver`** claim: SciTokens uses **`ver`**, WLCG uses **`wlcg.ver`**

Some technical points 2/2

- Groups claims:
  - Used for different purposes: both for authorization and accounting/defining the context/namespace:
    - Different AuthZ schemes don't mix well
    - Need for accounting etc.
  - Differences between WLCG and AARC
    - AARC values globally unique but deemed too long
    - WLCG short but risk of collisions
    - ↠ ideas on both sides how to unify/simplify
  - Claim name: perhaps **`groups`** (from RFC9068) ? Problem with too restrictive format
- Need for a namespace and/or VO to set the context of claim values, solutions related to groups issue above:

  see GUT profile running notes for the current discussion on this

Future

- Definitely long way to go
- Format seems to work: good active discussions
- We can certainly use more experts and help with organising

Thoughts, questions?