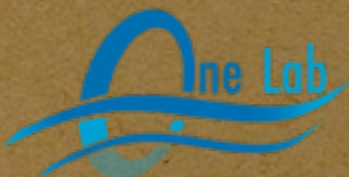


# Feedback on SFA API

Thierry Parmentelat  
INRIA - Sophia Antipolis

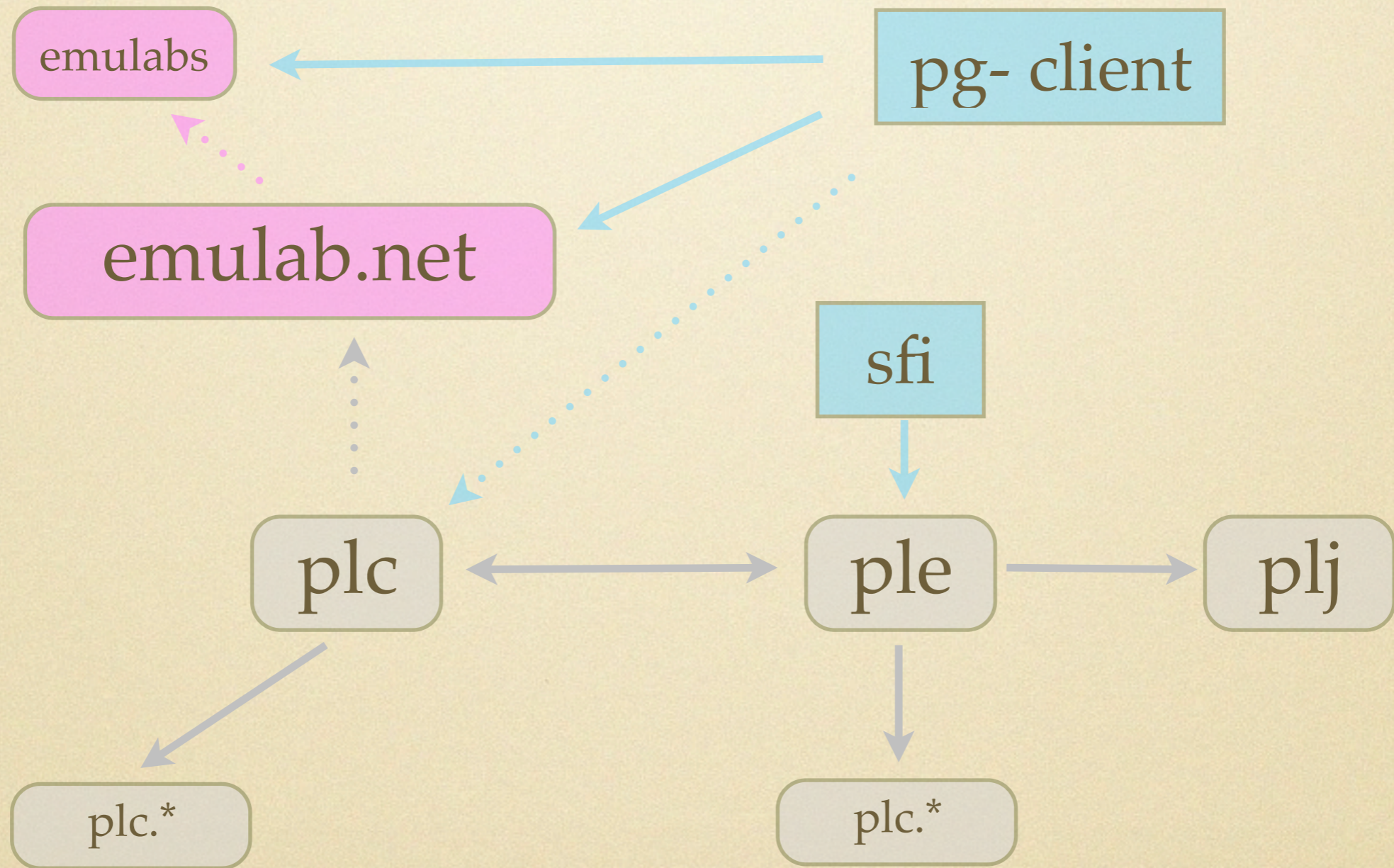


GEC9 - Nov. 3, 2010

# Challenges

- Get our users to use the SFA API
- Provide something better than current MyPLC
  - either by offering more possibilities
  - and / or by providing at least as good tools
  - easy discovery, WebUI
- Avoid to have users to find out the entry point

# Current setup

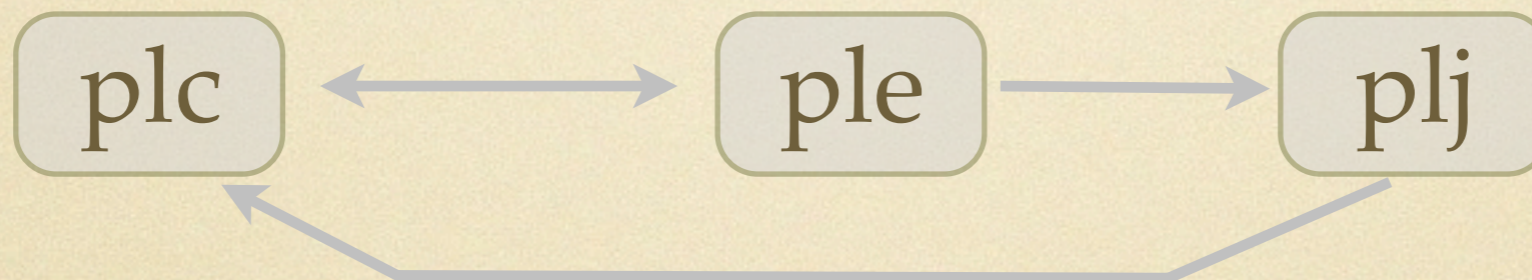


# Configuring MyPLC's SM

- MyPLC-based SFA SM service has 'neighbors'
  - pointing to SMs
  - or to AMs
- for overall scalability, preferred is SMs
  - using AMs is like 1-hop connectivity

# Anti-loop

- adding one peering link to the current setup



- triggers infinite loop

# Reliable tracing

- remote MyPLC not aware of who actually issued call
  - user (sfi) -> PLE (user's credential)
  - PLE -> PLC (PLE's credential)
- not sure how right this is
- “delegation” mechanism to address this issue
  - poorly specified

# Proposed extension

- for both:
  - effective anti-loop
  - reliable tracing
- propose to keep track of the call chain

# Probing the federation

- nice to have federation introspection
  - federation scope (spot services)
  - assess versioning (flavour(s), sw)
  - and related configuration
    - trusted roots
    - configured neighbors (PLC flavour only ?)



# How to probe

- each service to expose its neighbors
- allows for probing federation scope
  - transitive closure of 'neighbors' relationship
  - from anywhere (does not require 'root')
  - side benefit: check local auth. scope
- in PG, relates to ListComponents ?
- currently using 'GetVersion'

# Resource discovery

- Need for filtering capabilities
- can be done in rspecs
- might need more builtin mechanisms though
  - e.g. if that affects routing calls

# rspec good practices

- rspec : testbed-specific
- testbeds could expose local jargon (e.g. xsd)
  - as part of probing method
- such formalisms would greatly help
  - in writing more helpful user-tools
- how to share / re-use for similar topics

# Evolution process

- SFA-2.0 ?
  - supporting several authentication schemes
  - certainly is doable
  - but comes at a cost in terms of complexity
  - what does that buy us in return ?
- more generally, would like to contribute more

# SFA-enabled web UI

## Key requirements for an SFA Client

- User's Private Key
- User's (Generated) Certificate
- Storage area for Credentials
- XML-RPC

# SFA-enabled web UI

## Key requirements for an SFA Client

- User's Private Key  
Client Certificate (pkcs12)
- User's (Generated) Certificate
- HTML5 (localStorage, File Access)
- XMLHTTPRequest