Security on the Worker Node

A jump start

TEG F2F Amsterdam
26th January 2012

steffen.schreiner@cern.ch

In a nutshell

I. Help preventing security incidents from spreading or reoccurring

II. Ensure compliance with legal requirements, including due diligence

III. Provide deniability for users who were not involved with an incident. Effectively, one would like some serious proof of who did (or did not) what.

In the short run

I. **Help preventing security incidents from spreading or reoccurring**

II. Ensure compliance with legal requirements, including **due diligence**

III. Provide deniability for users who were not involved with an incident. Effectively, one would like some serious proof of who did (or did not) what.

Where we need to get now

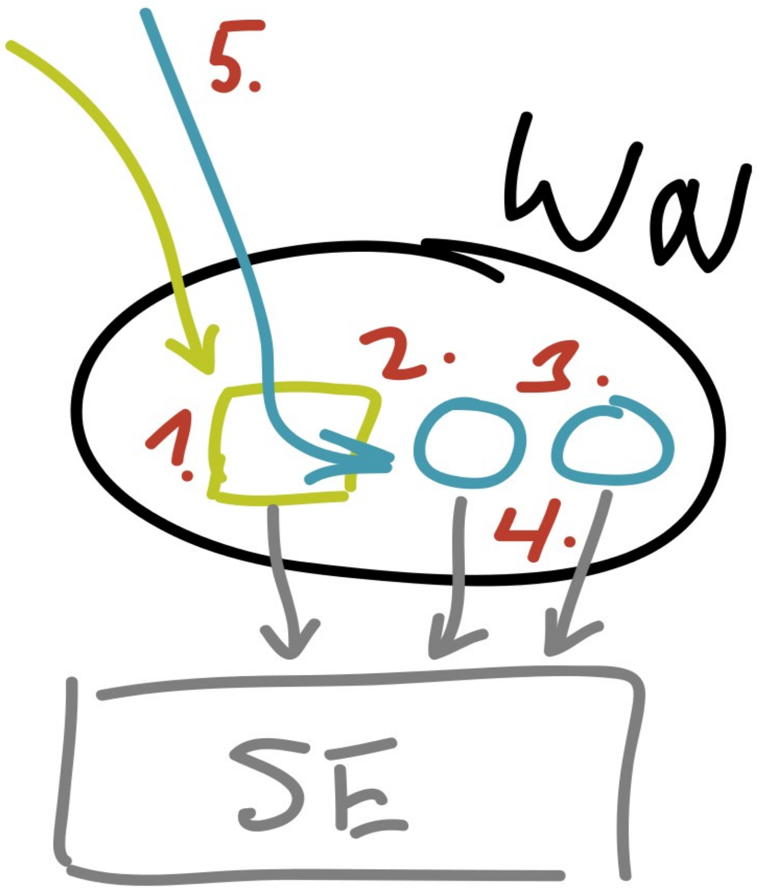
- **Agree on** and specify **solution**(s) to **protect the Pilot** and **isolate** Grid jobs immediately
- Specify a **trust model** that defines what is necessary to assure in the future
- (Even with no potential solution at hand) We need to agree on and state what are the problems to solve

Just a short reminder...

- **Proxies** are not a clever way of proofing a Grid job's authenticity on a WN:
  - **Can be exchanged**
  - **Can be newly applied (to jobs)**
  - **Don't say anything about a job itself**
  - **Very dangerous, allow identity forgery**
- Basic Argument is: If you have all the fuss to get the credential to the WN, then you should demand much more from the mechanisms than what a proxy could proof

Technical questions

1. How to **limit** the Pilot's credential
2. How to **protect the Pilot** and its **credential** from Grid jobs
3. How to **isolate Grid jobs** from each other
4. How give the Grid job a **credential** for IO access
5. How to proof a Grid job's authenticity



First Step (short term)

- Protect the Pilot (id switch)
- Isolate the jobs

We need to pick:

- gLExec with a string provided DN ?
- Sudo (no id management, no env. Switch)
- SELinux, cool thing, but a long way to go
- Virtualization – maybe not for everybody, but maybe as a another option!?

Trust model

First question we need to answer:

“Is a VO and its users one trust domain?”

My opinion is no, because...

- Don't think in a mix-up of roles. Users are generally not trusted, even if some guys are also developer or administrators.
- If VO+users are one trust domain/entity, then why are there quotas and policies?
- We cannot trust a user's behavior regarding his system and credentials

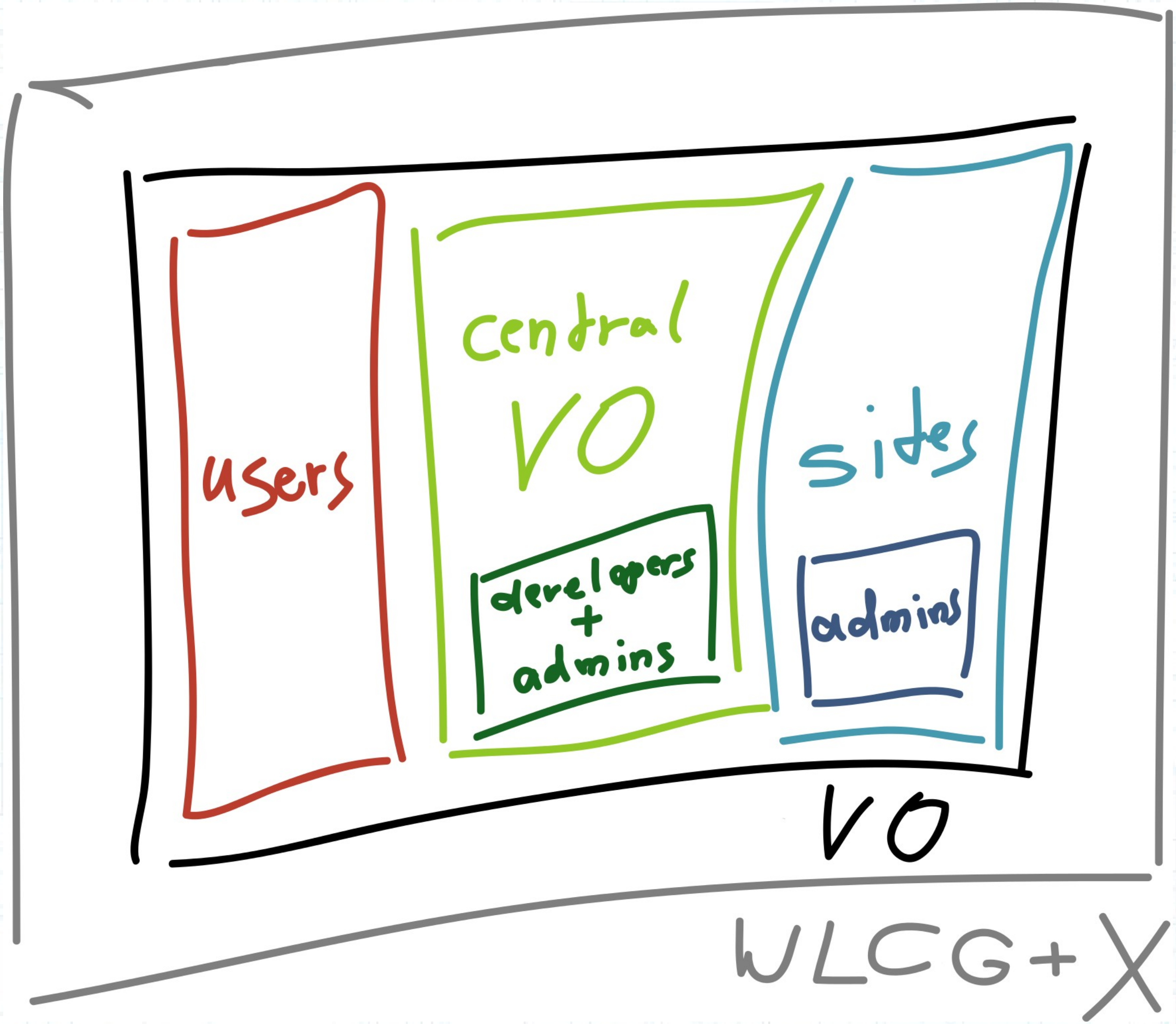
Trust model – why I.

- What is proven by a user proxy certificate to a site upon job execution?
- Can a VO be trusted to provide the right payload and digital identity for MUPJ ?
- If yes, then we don't need user proxy certificates here.
- If no, then we need a new way of delegating Grid jobs. User proxy certificates are no good at all to prove accountability/traceability. This is detailed below.

Trust model – why II.

- Do we need to protect a VO and its admins from potentially using or leaking its Grid users credentials? Do we need to provide for plausible deniability in case of incidents?
- If yes, then user proxy certificates may not leave the user's computer.
- If no, then we maybe should not worry about trust in the VO concerning traceability and logging on the WN.

Trust model / players
a suggestion to start with



Trust model – getting there...

- Once we have identified the different trust domains or players, we need to make clear what's their relation (at least roughly)
- We will see that we want to ensure some conditions between these players
- This should provide us a clear result what are the consequences for the accountability on a WN
- Thereafter, we can define what we (want) need to proof to whom on the WN and before

Let's ...

- stick to a though timing schedule!

- not talk about products or solutions where its still not formulated or agreed on what is the problem we want to solve!

Thanks! =)

Schedule

I. Intro, questions, getting on speed (15min)

II. Short term, id switching (30 min)

   I. SELinux
   II. sudo
   III. gLExec
   IV. Virtualization
   V. Summary and stated conclusions

III. Trust modeling (45 min)

   I. Trust domains and players
   II. Player's relations
   III. Consequences and arising problems
   IV. Summary and stated conclusions