



**RIPE
NCC**

Personalised Authorisation

Tim Bruijnzeels

Assistant Manager Database Group

- Maintainers and Persons are hard!
 - Lots of support requests
 - Maintainer reset process
 - People confused at training courses

- User experience and tools suffer
 - Complicated object creation and maintenance
 - Complicated set up for new LIRs
 - Complicated authorization management

- WG feedback needed!


- Person objects MUST be maintained
 - So we recommend people to use their own maintainer
 - And since this is cumbersome to set up, we have a tool <https://apps.db.ripe.net/startup/>
- We also have SSO accounts
 - Another (private) identity for users to maintain
 - Used in more and more places
 - Provides authentication (two-factor optional)
 - Easy password recovery

Manage your RIPE NCC Access account.
You can make changes to your RIPE NCC Access account at any time. Update your password or your personal information here.

Services that you are subscribed to:

- > SSO Default
- > RIPE Labs
- > LIR Portal

Edit your subscriptions




Welcome, Tim Bruijnzeels (sign out)

Your email address
tim@ripe.net
[Change address](#)

Your name
First name: Tim
Last name: Bruijnzeels
[Change name](#)

Your password
Current password:
New password (minimum length: 8):
Confirm new password:
[Change password](#)

Two-step verification
Status: Off 
[Set up two-step verification](#)

```
mntner:      tim-br-mnt
descr:      Startup maintainer
admin-c:    TB7733-RIPE
upd-to:     tim@ripe.net
auth:      MD5-PW # Filtered
mnt-by:    tim-br-mnt
referral-by: tim-br-mnt
changed:   tim@ripe.net 20141024
source:    RIPE # Filtered
```



```
person:      Tim Bruijnzeels
address:     Singel 258
             1016 AB Amsterdam
             Netherlands
phone:      +31205354444
e-mail:     tim@ripe.net
nic-hdl:    TB7733-RIPE
mnt-by:     TB7733-RIPE
auth:      SSO
auth:      PGP
changed:   tim@ripe.net 20141024
source:     RIPE
```

- Maintainers are anonymous

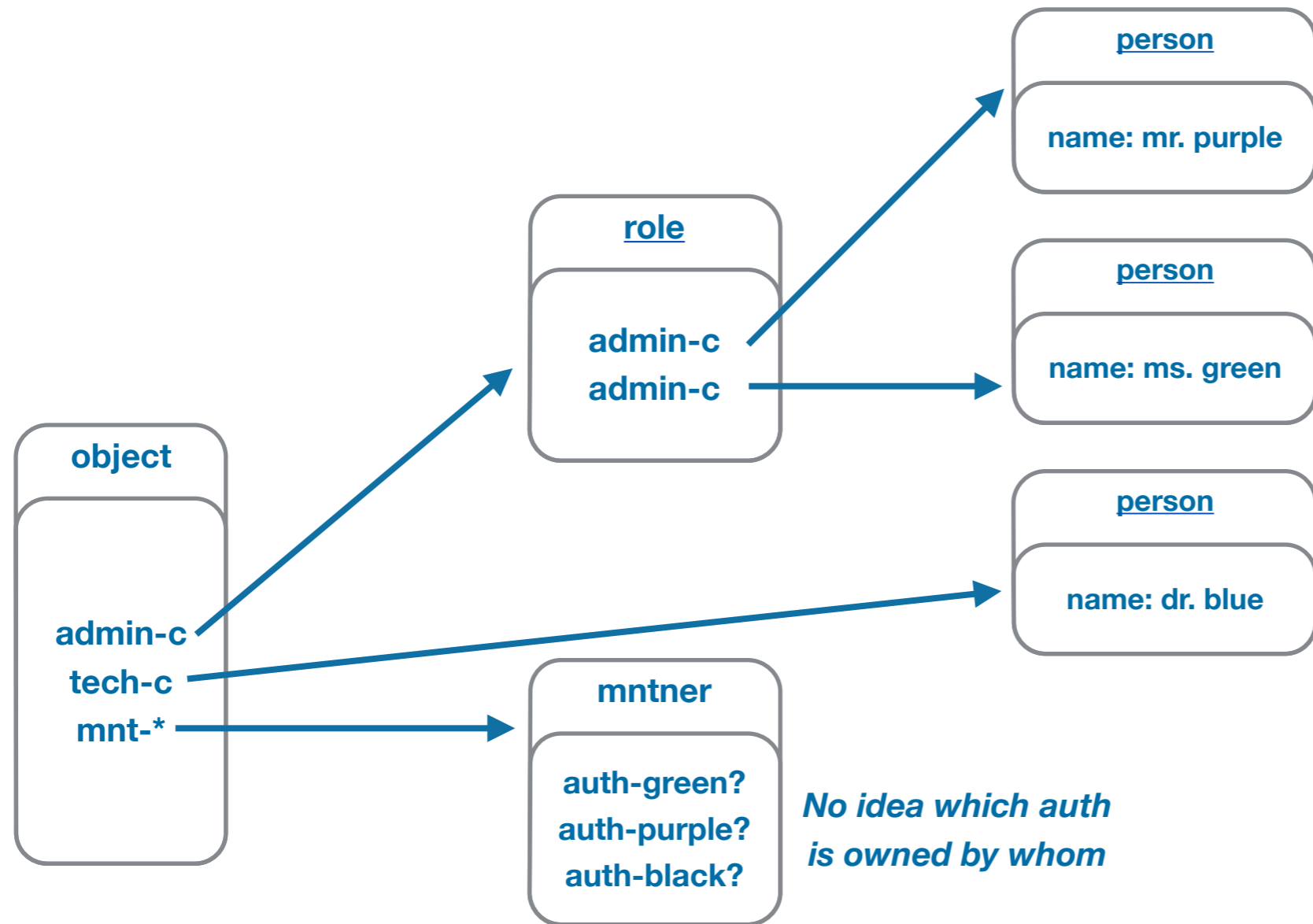
- Complicates recover access to a maintainer if password is lost
- Updates are accepted as long the authentication provided matches any of the authentication values, on any of the maintainers

Complicates showing sensible update logs to **authorized** users

- Maintainers are not intuitive

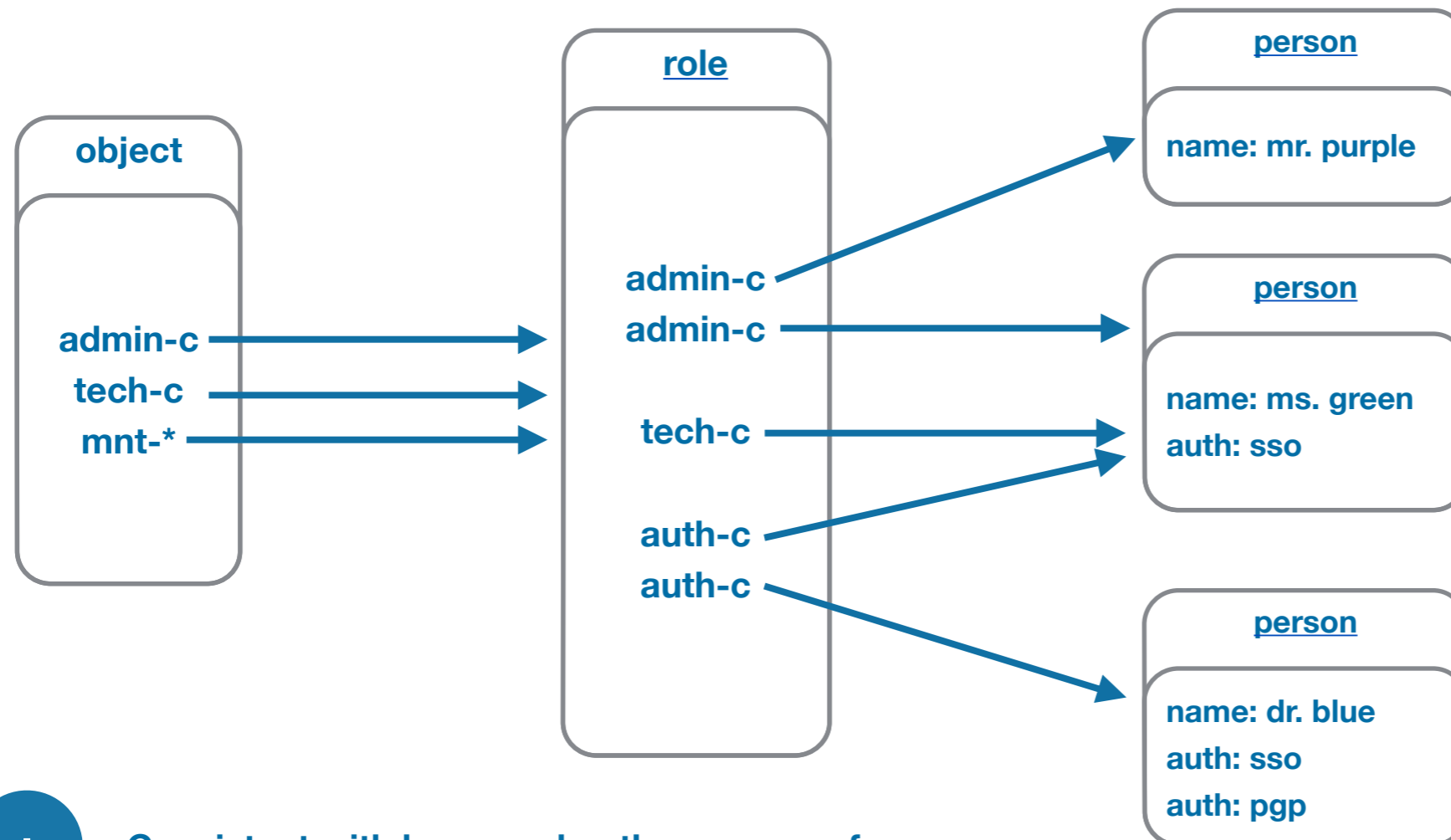
- Contrary to other RIPE DB objects it is not clear what exactly a maintainer represents (person, organisation, group, script)
- Most security systems use an approach that separates **authentication** (who you are), from **authorization** (what you are allowed to do), and groups individuals in **roles**:

My colleagues and I (*authenticated* persons) are part of a group (role) that is authorised to maintain this object (mnt-*)

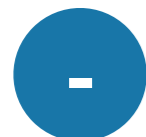


direct reference to person without role discouraged because of poor scaling

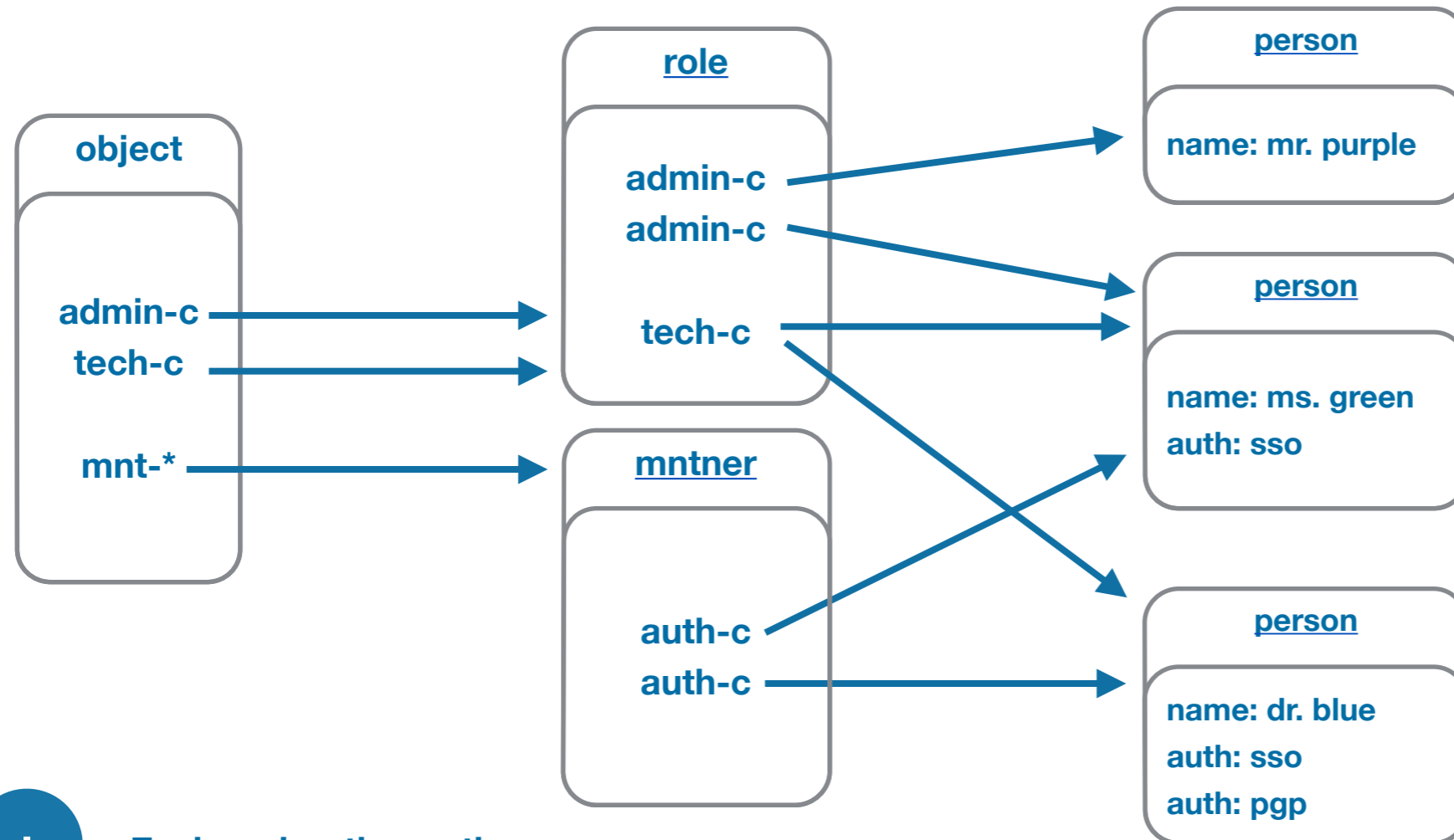
No idea which auth is owned by whom



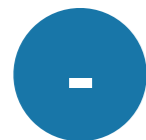
Consistent with how we do other groups of persons



More difficult migration path (will get back on this)



Easier migration path

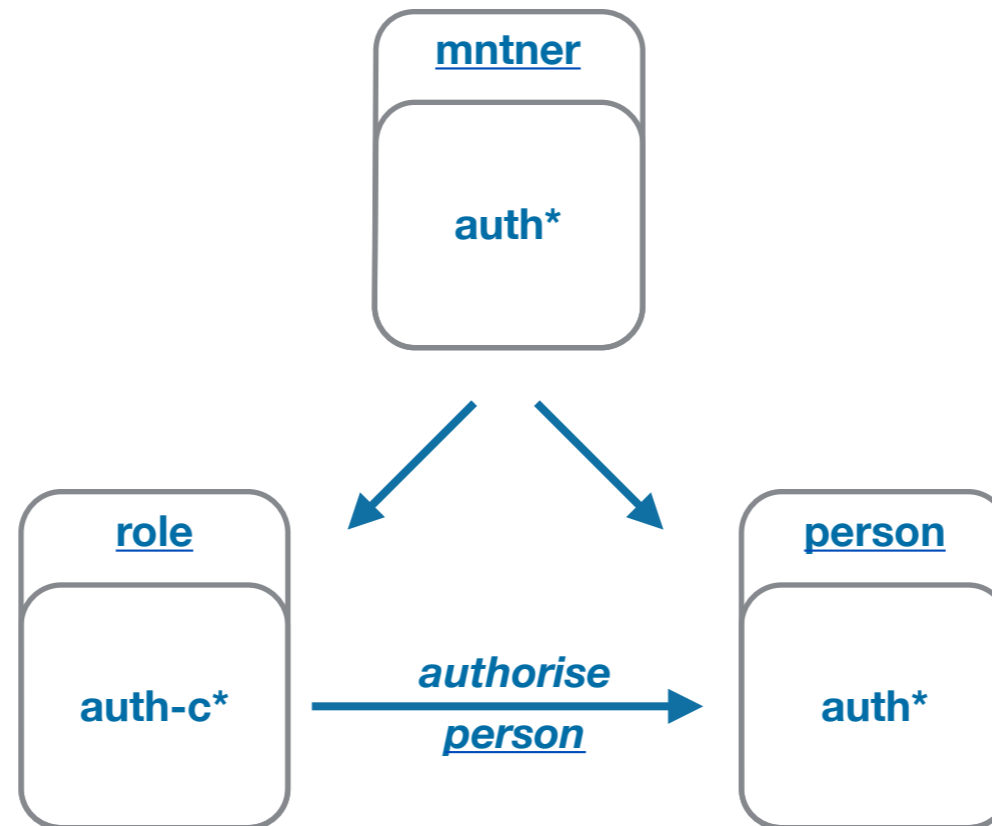


Having roles for groups of contact persons, and mntner for groups of authorised persons may be confusing

- Keep different mechanisms?
 - Adds complexity and confusion
- Create new, and delete old some time later?
 - We have 50k maintainers.. there is no way we will get everyone to update
- Should remain compatible at all times
 - No action should be required from current users
 - I.e. any forced migration should be fully automatable
 - Everything should keep working
 - Improvements available to new and existing users

- Updates include credentials (password, pgp, etc), but do not mention which **maintainer** is supposed to contains them
- The software checks all eligible **maintainers** for a match
- The software can still easily find matches by checking **roles** and **persons**, or **persons** on **maintainers**, instead of just **maintainers**

In short: nothing needs to change here..



Convert most of the **mntner** into a **role** with the same name

Move all auth lines into a single anonymous **person**

But how do we deal with all the attributes?

attr	mandatory or optional?	problem?
mntner	lookup key (name)	364 out of 50055 clashes
descr	mandatory	make optional in role
upd-to	mandatory	make required when role is used to maintain
mnt-nfy	optional	make optional when role is used to maintain
auth	mandatory	use auth-c in role and allow auth in person
referral-by	mandatory	will be removed

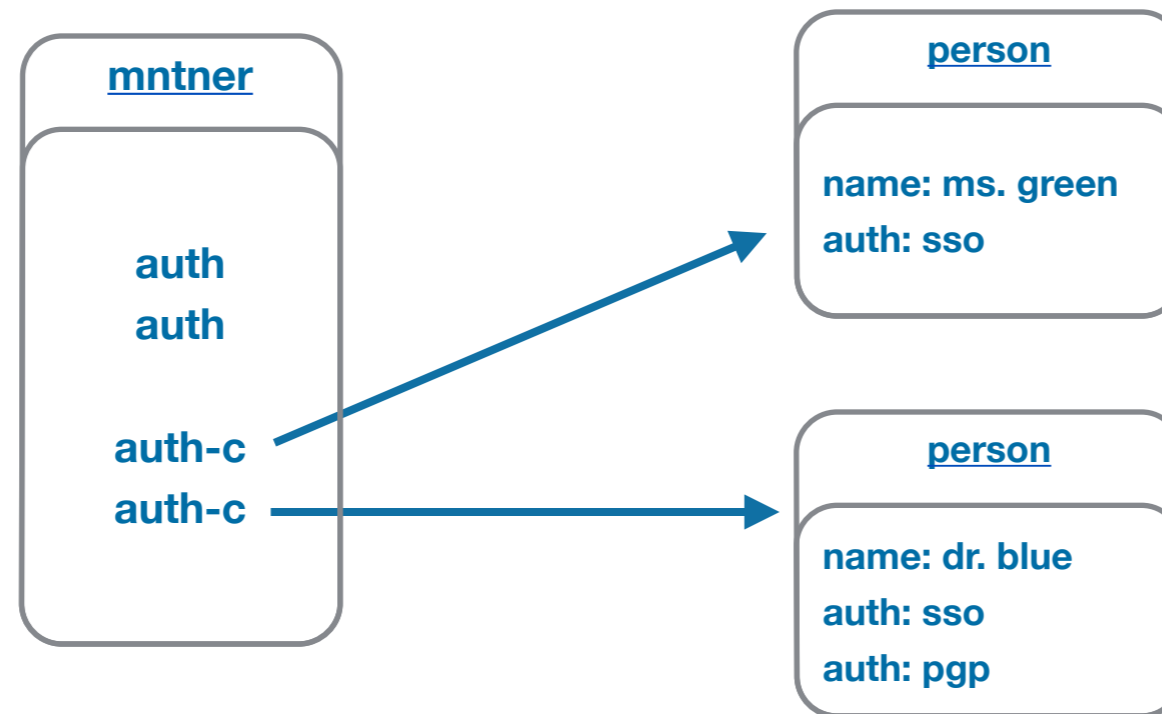
- i) Should be possible to fix case by case.
- ii) This might be useful in roles anyway.
- iii) We will need these anyway if a role is used in a maintainer context. And we can have business rules to ensure that they are set when needed.
- iv) Implement business rules to require that auth-c is present when using a role in a maintainer context, and that auth is present in persons referenced.

Filter “auth-c:” from normal output! Similar to “auth:” now, this is nobody’s business but yours.

attr	mandatory or optional?	problem?
address	mandatory	use 'unknown'? make 'optional'?
phone	optional	leave blank
fax-no	optional	leave blank
e-mail	mandatory	use 'upd-to'? 'unknown'? make 'optional'?
nic-hdl	mandatory	auto-generate

attr	problem?
person	generate anonymous-person-for-roleX
address	use 'not applicable'
phone	use 'not applicable'
nic-hdl	auto-generate
mnt-by	allow self maintaining
remarks	anonymous person object for role X, see: http://www.ripe.net/.../documentation/...

- Filter from normal queries? This is not of general interest.. authorized users that know the “auth-c:” in the role, can find it by nic-hdl.
- Do not allow updates? Encourage setting up proper persons (with tooling)?
- Or is there a more general use? A ‘person’ object for your organisation, to support automated tools?



- No migration needed for current maintainers
- No anonymous persons, maintainer is still the logical point for scripts
- But allows referring to natural persons where possible
- Allows easy automatic migration of current “auth: sso” to “auth-c: person” once the person has its own “auth: sso”

- WG discusses
- Consensus
- RIPE NCC provides implementation plan
- WG discusses
- Consensus
- RIPE NCC implements

