

Wie validiert man eine solche Kette?  
 Algorithmus, der diese Frage beantwortet.

Eingaben:

- Zert-Pfad
- gegenwärtiger Zeitpunkt
- zulässige Policies  
(akzeptabel für Verifizierer)
- any Policy möglich (wenn Policy egal)
- trust anchor (selbst-signiertes Zert)
- initial policy mapping inhibit true / false
- " explicit policy true / false
- " any policy inhibit true / false



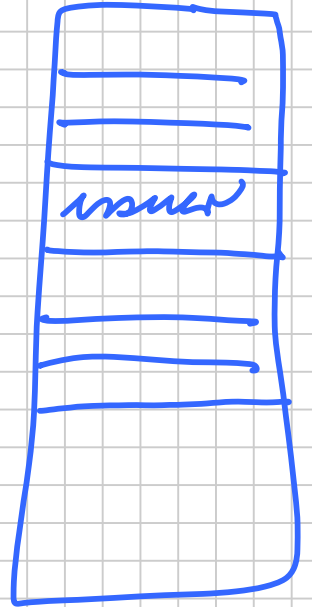
working public key      aus trust anchor

...      param.      aus trust anchor

issuer      aus trust anchor

max path length: initialisiert mit  
 $n = \text{Zertpfadlänge}$ .

Processing  $i = 1, 2, \dots, n$



- verify basic cert info

- issuer  $\stackrel{?}{=}$  working issuer

- cert signed with working  
pkalg, working pk, working pk param  
verify sign.

- actual date in validity period?

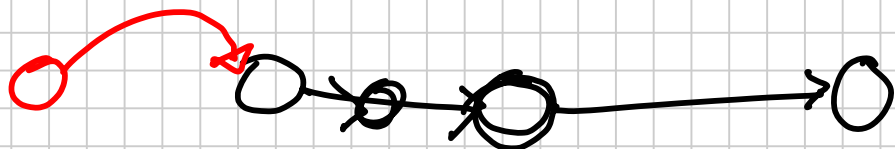
- cert revoked?

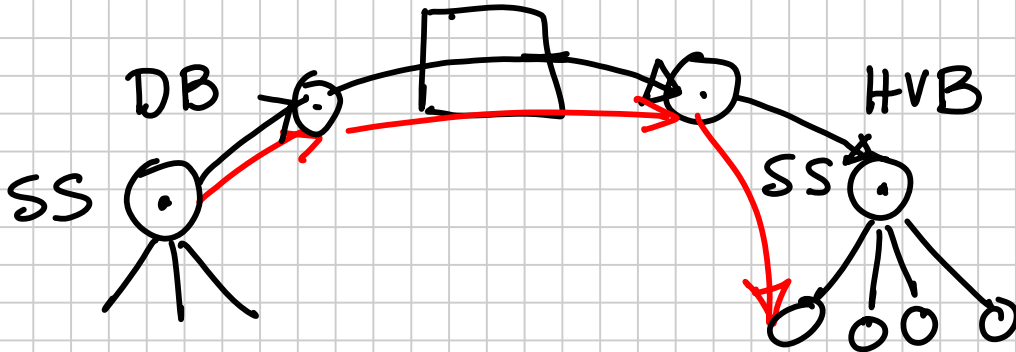
- extensions verständlich?

- verify name constraints. (nur  
für nicht selbst. signierte Zert).

Beispiel für selbst-sign. Zert. in

Pfad      DB      Bridge      HVB      Kunde





## Policy information processing

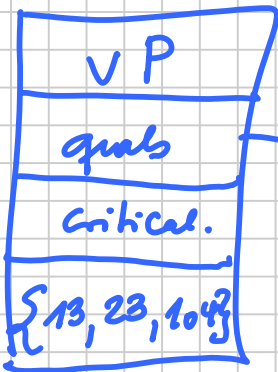
23

Für jede P in

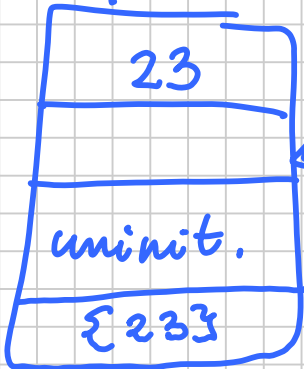
pol. set

Schau an alle Knoten in Policy tree der Tiefe  $i-1$ .

policy set → Policy ext  
23, 97, 103



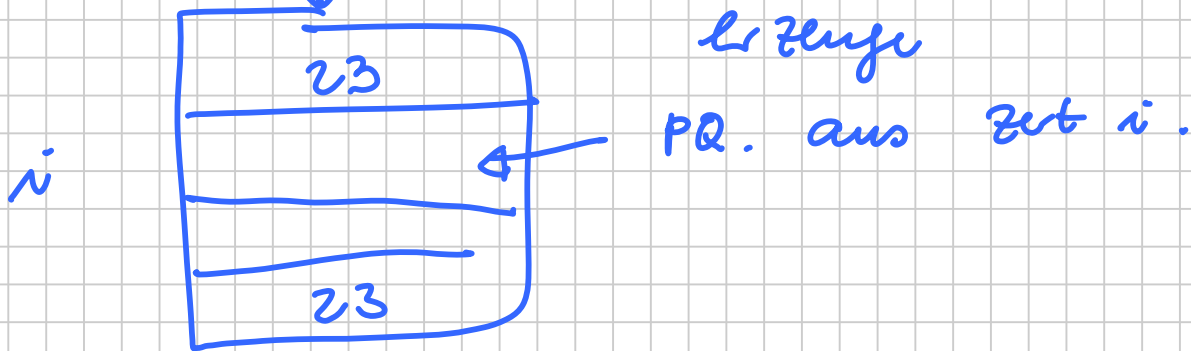
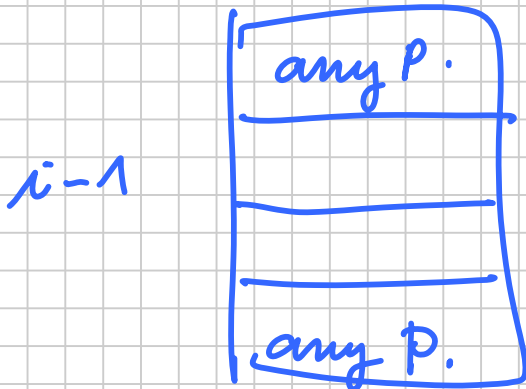
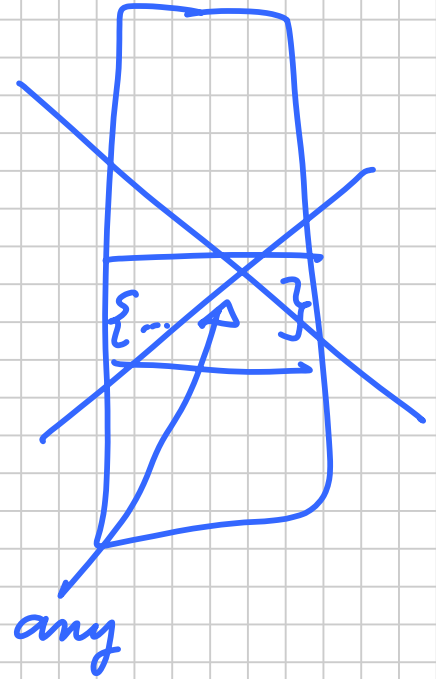
Finde einen, in dem P in exp. policy set vorkommt.



"übernehmen"

wenn P (23)  
nicht gefunden  
wird,

Wenn aber ein  
Knoten der Tiefe  $i-1$   
gefunden wird



any pol. kommt in  
pol. ext. vor.

ist any policy auf level  
 $i$  erlaubt?

inhibit any pol.  $> 0$  ?  
oder  $i < n$  und Zert  
is self issued.

