




Riverbed


Enforcing User-defined Privacy Constraints in
Distributed Web Services


Frank Wang, MIT CSAIL; Ronny Ko and James
Mickens, Harvard University





What is it?


- Platform that simplifies the creation of web services that respect user-defined privacy policies.
 - Uses simple policy language
 - Works with unmodified, legacy software
 - Does not require developers to manually annotate code with labels
- 



What problems does it address?

- LOSS of the user

- Enforce w/ runtime






Riverbed's solution

For developers:

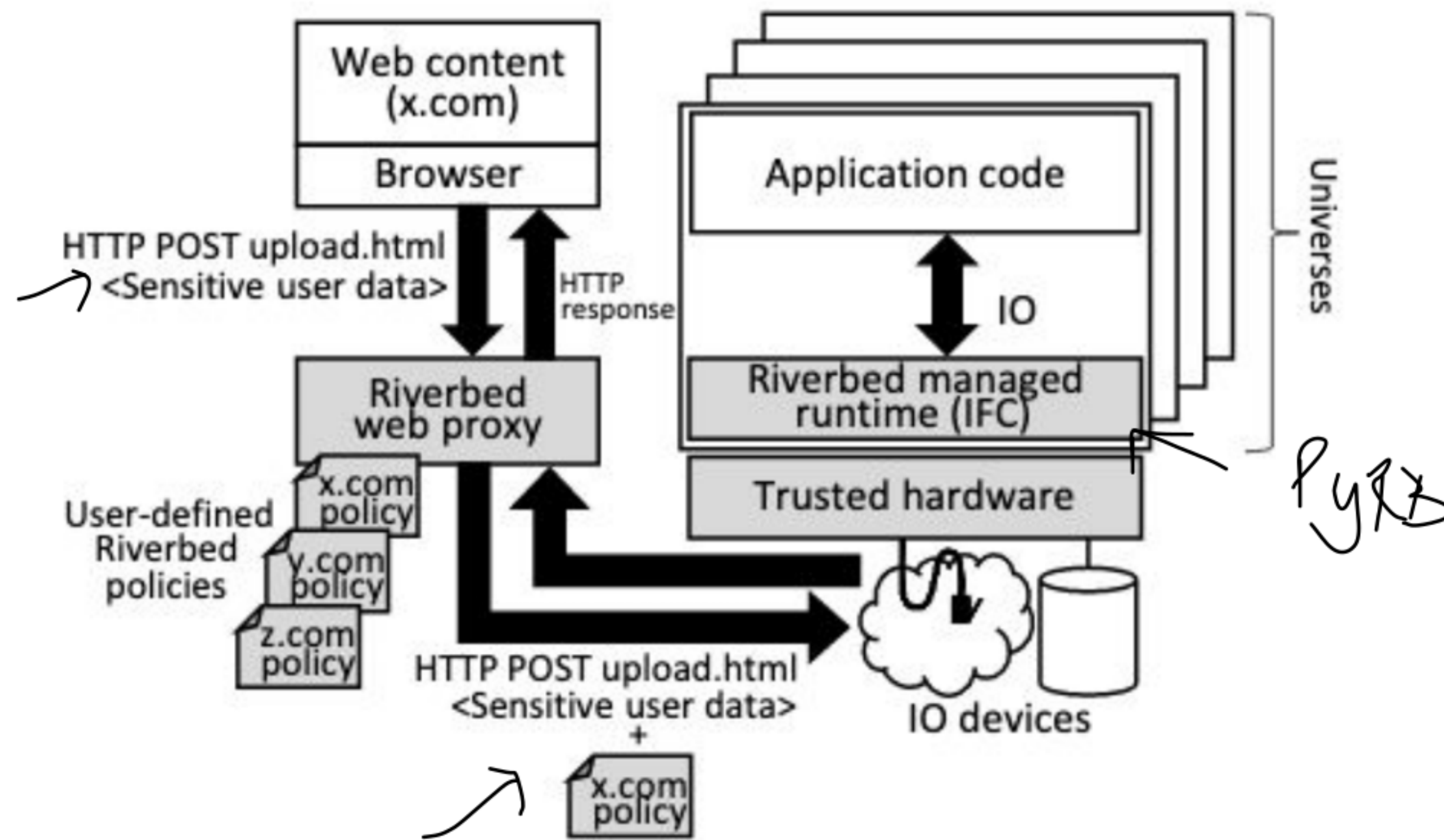
- practical
- RB on top of existing complex applications to enforce stronger security policies

For users:

- straightforward way to define their own policies
 - ensure that server-side code is preserving their privacy
- 

Overview of RB's Design



A: "don't store my data to persistent storage"



What are the consequences of a user defining some incompatible data flow policy?

Eg. Dropbox :


- crashes
- the server!




What are the consequences of a user defining
some incompatible data flow policy?

Riverbed would terminate the application.

What does that mean for everyone else using that application?



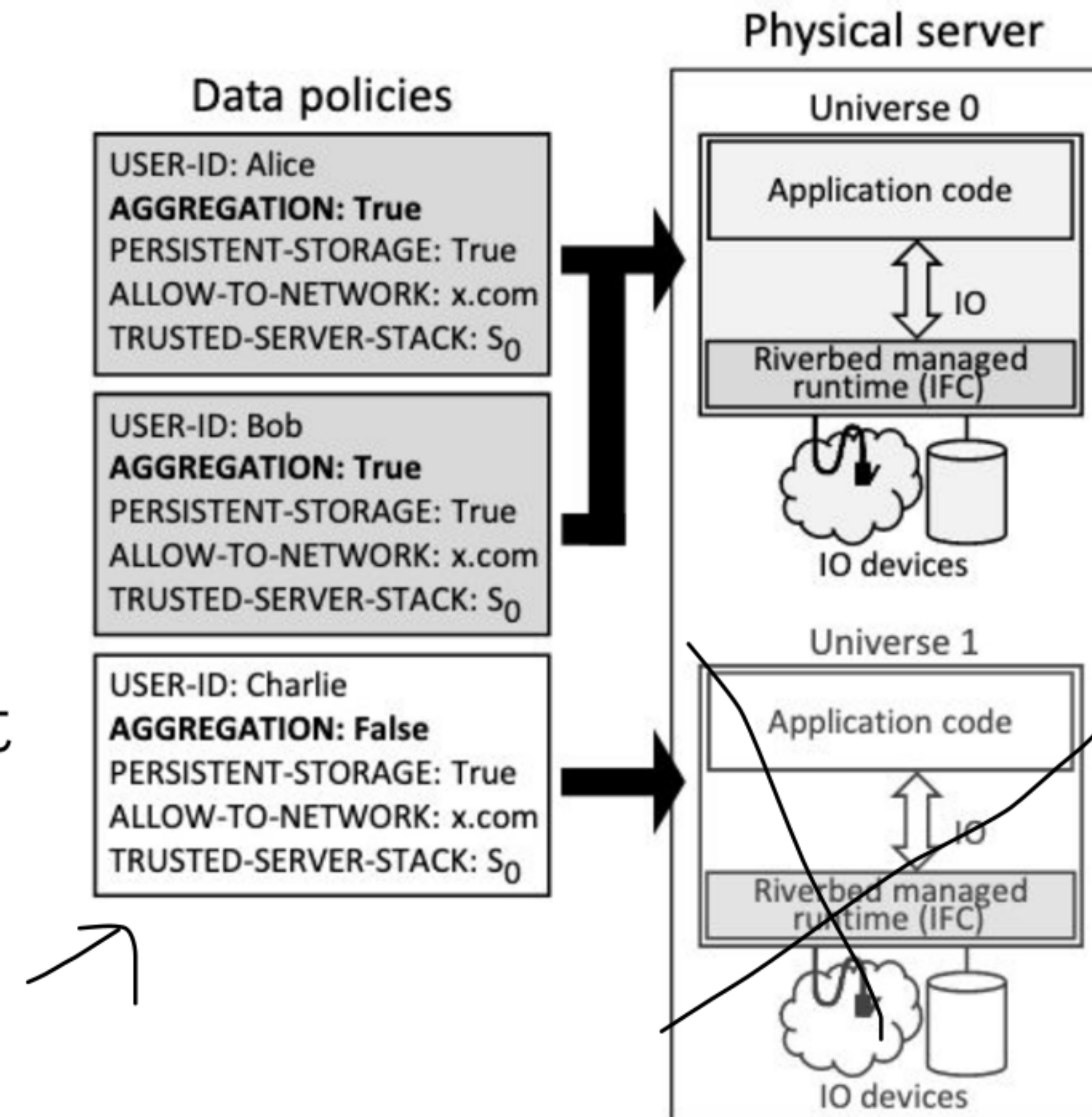
What are the consequences of a user defining
some incompatible data flow policy?



Service is unusable for everyone because Riverbed
terminated the application!!

Universes! :O

- RB forks the application and creates a separate universe for each set of users with the same data flow policy.
- My incompatible data policy won't affect others.





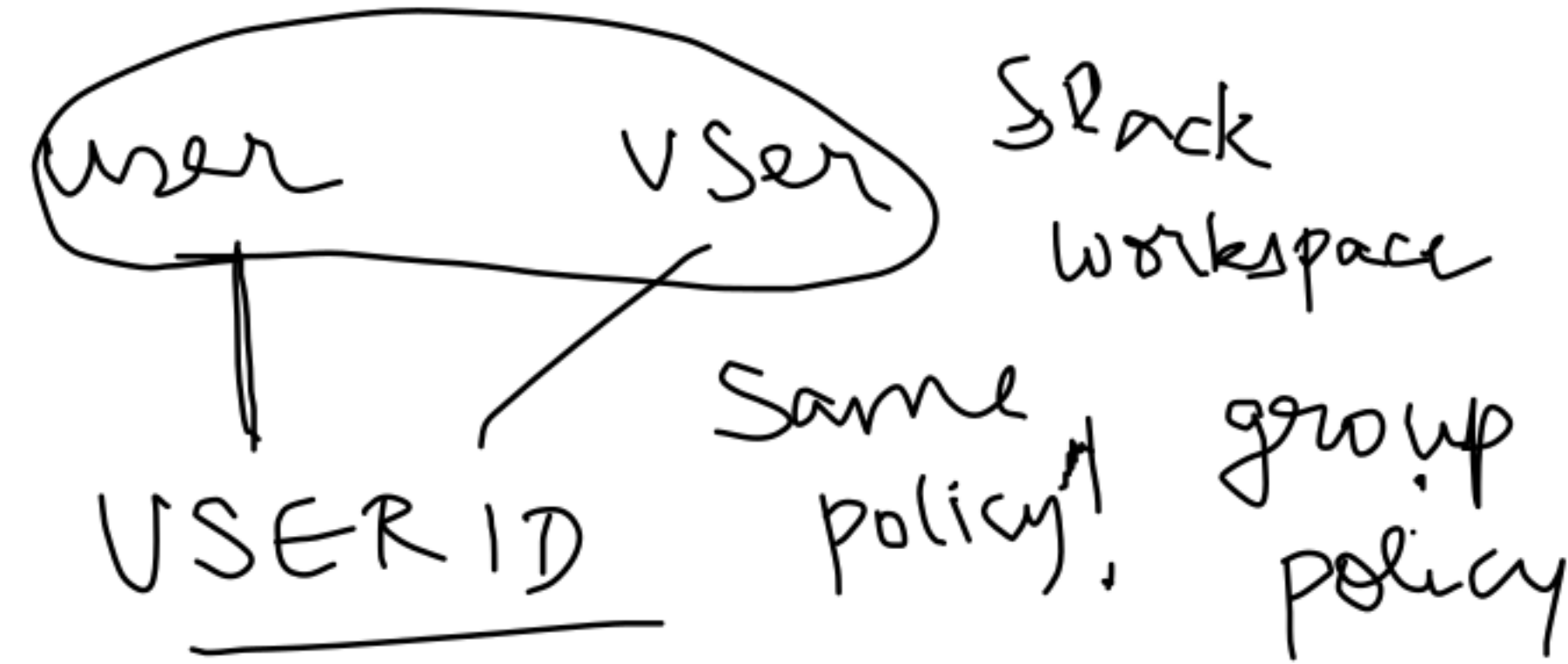
Remote Attestation




- Need to ensure that the server is running an IFC-enforcing Riverbed runtime
- Proxy on the user's end needs to verify this
- More on this later

Policies


```
USER-ID: ALICE
AGGREGATION: False
PERSISTENT-STORAGE: True
ALLOW-TO-NETWORK: x.com
ALLOW-TO-NETWORK: y.com
TRUSTED-SERVER-STACK: {
  83145c082bbf608989f05e85c3c211f83,
  c8cd7ac93cab2b94f65a5b2de5709767f,
  ...
  590f01d8d18b1141988ee1975b3ce3b30
}
```




yes/no add?
 count my votes
 reddit.com / ... specific
 aggregation:




Can I disallow overall aggregation but allow specific types of useful aggregation?

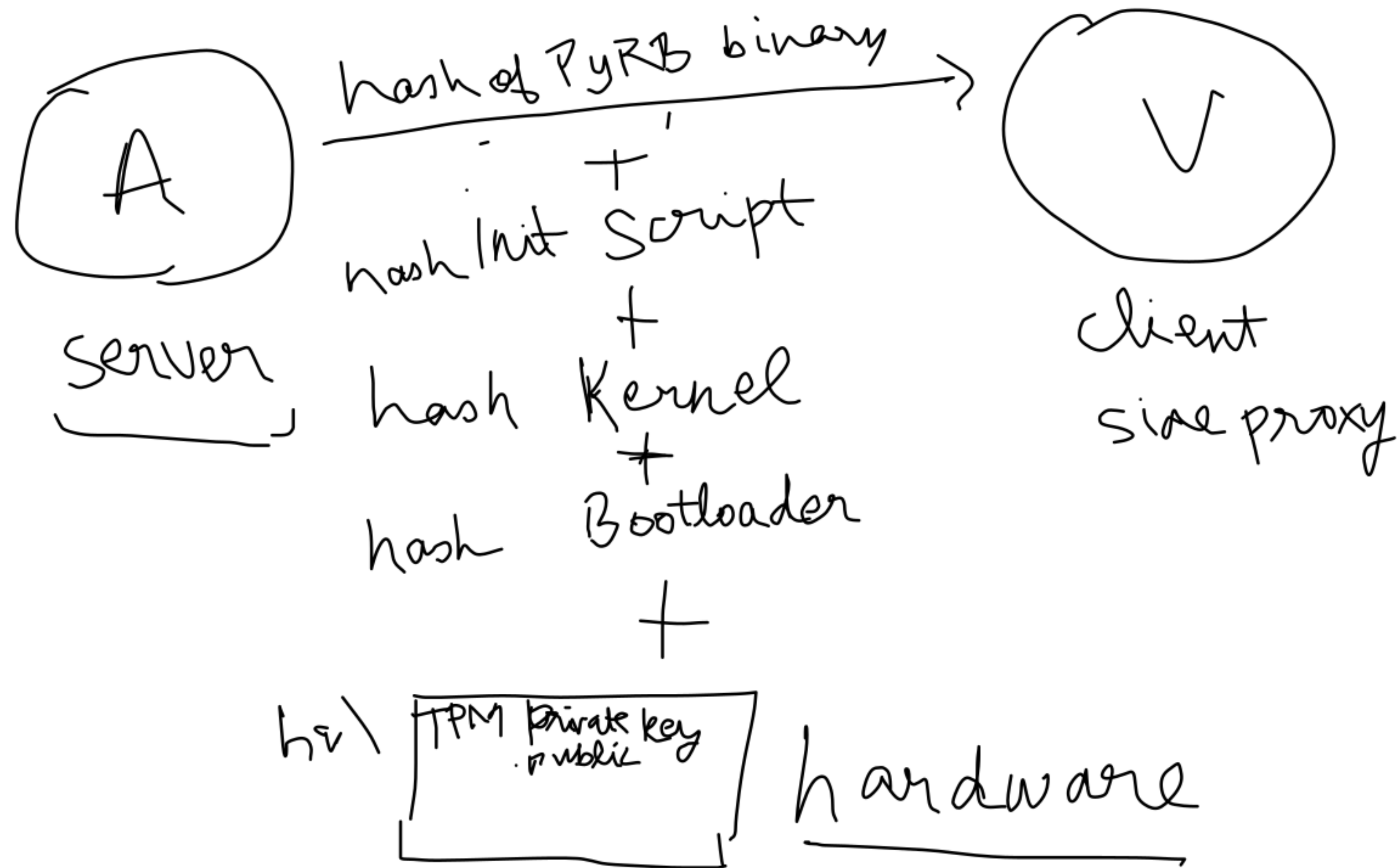




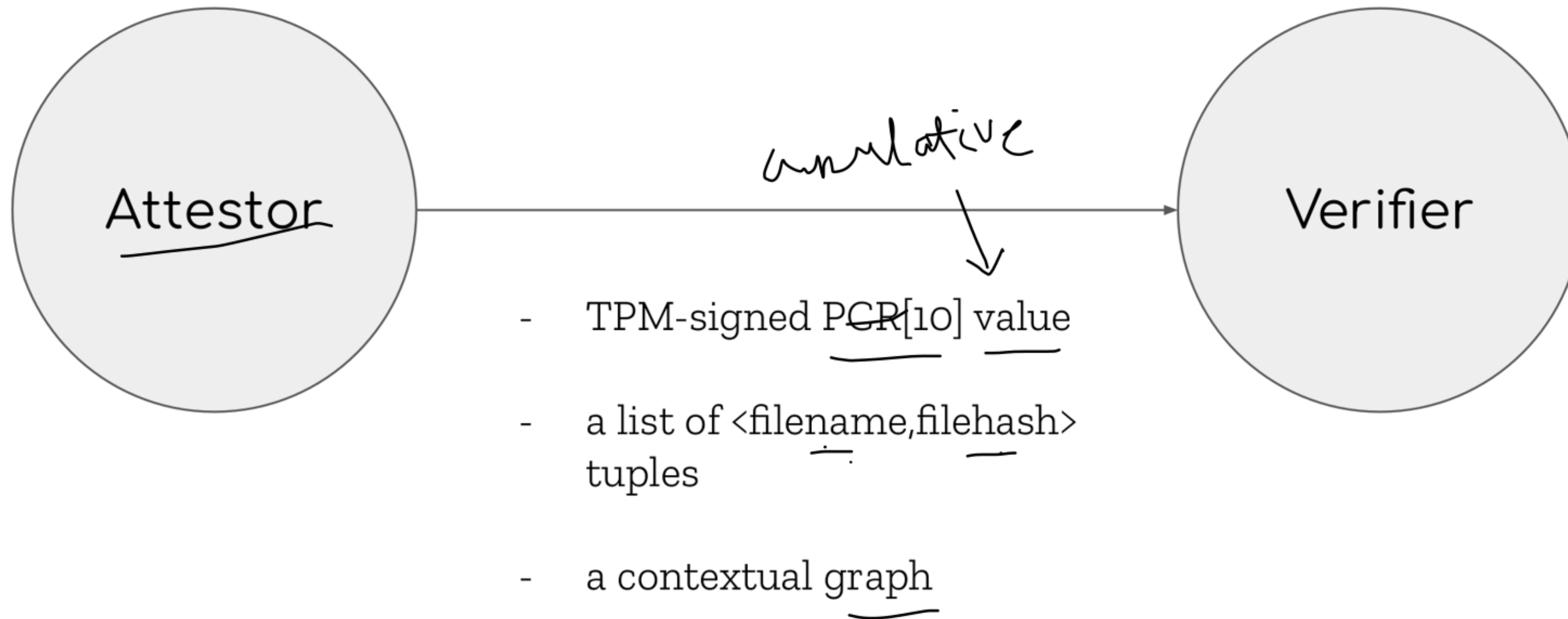
Can I disallow overall aggregation but allow specific types of useful aggregation?

- Yes!
 - ALLOW-TO-NETWORK can have a child policy
 - For a specific endpoint, the child policy can override the parent policy, allowing aggregation at that endpoint.
- 

Server Attestation




Server Attestation





Discussion!

What are some strengths of Riverbed?

- maintain taint ~~at~~ universe level (or is it?)
 - Policies → user defined (nice :)
- 

Discussion!

DOOMED

What are some weaknesses of Riverbed?

- spatially inefficient
- app
- policies too restrictive
 - hence, not very expressive
 - if less restrictive, too many universes

A vertical column of 20 orange dots, arranged in 5 rows of 4 dots each, located on the left side of the slide.

Discussion!

Does Riverbed sufficiently address the problems it is trying to solve?

A vertical column of 20 purple dots, arranged in 5 rows of 4 dots each, located on the right side of the slide.



Discussion!




Do you think Riverbed can be adopted by large and complex web-services?

A vertical column of 20 orange dots on the left side of the slide.

Discussion!


Are there other solutions to make IFC
systems efficient other than Riverbed?

A vertical column of 20 purple dots on the right side of the slide.

A vertical column of 20 orange dots on the left side of the slide.

Discussion!

Are there other solutions to make IFC
systems efficient other than Riverbed?

A vertical column of 20 purple dots on the right side of the slide.

lets improve RB

- Policy should be on organization level
- Ship proxy with the app itself
- Prepackaged policy by trusted groups (eg. ACL^u,_{EFF})
- There are some good use cases (Slack)
- Resin + RB