Deferred Key Binding for OAuth Tokens

Justin Richer

Brian Campbell

Dean Saxe

OAuth WG IETF122

Trust me, bruh.

Justin Richer

Brian Campbell

Dean Saxe

OAuth WG IETF122

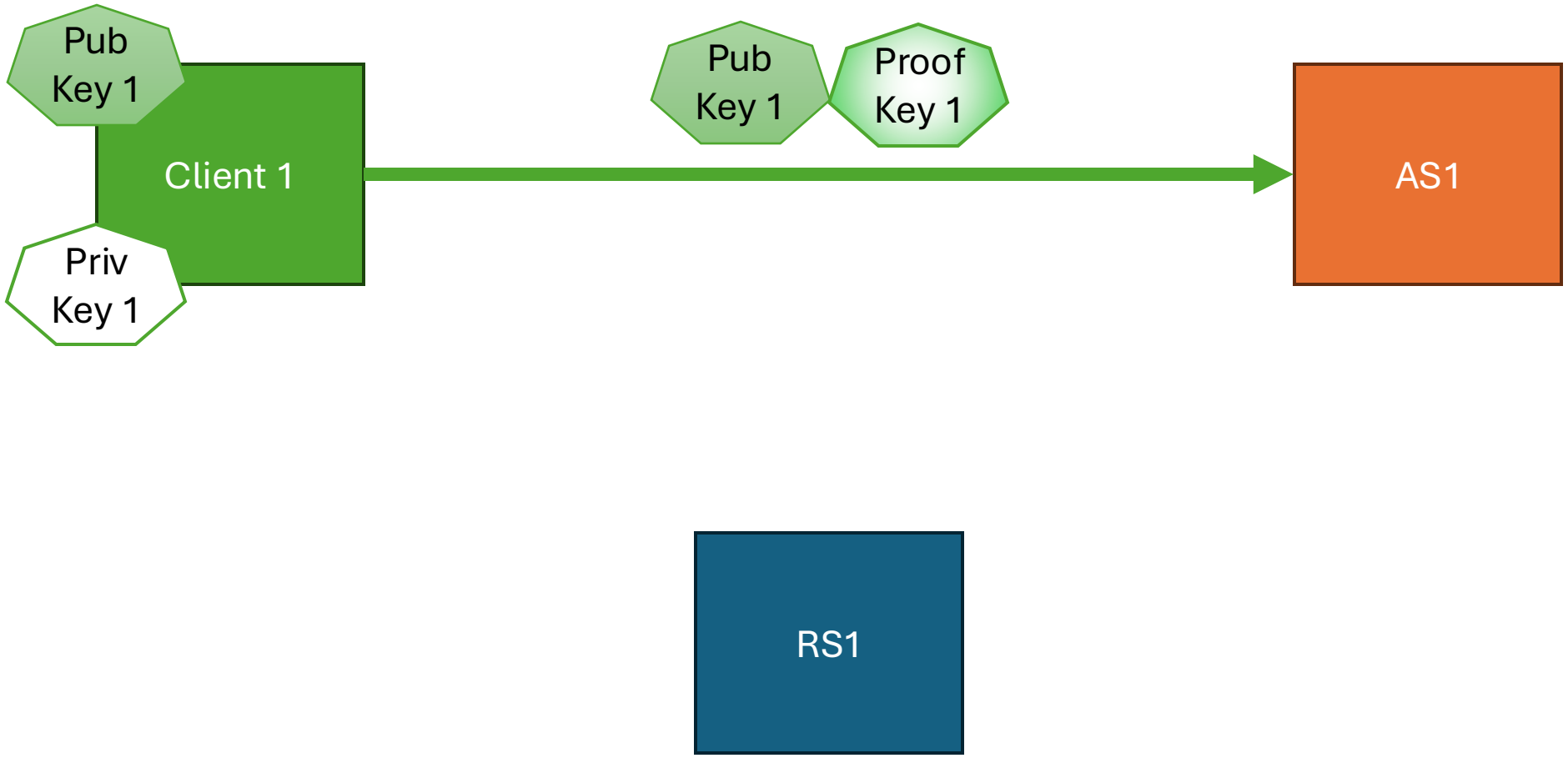
PKC In A Nutshell

Anyone can know this

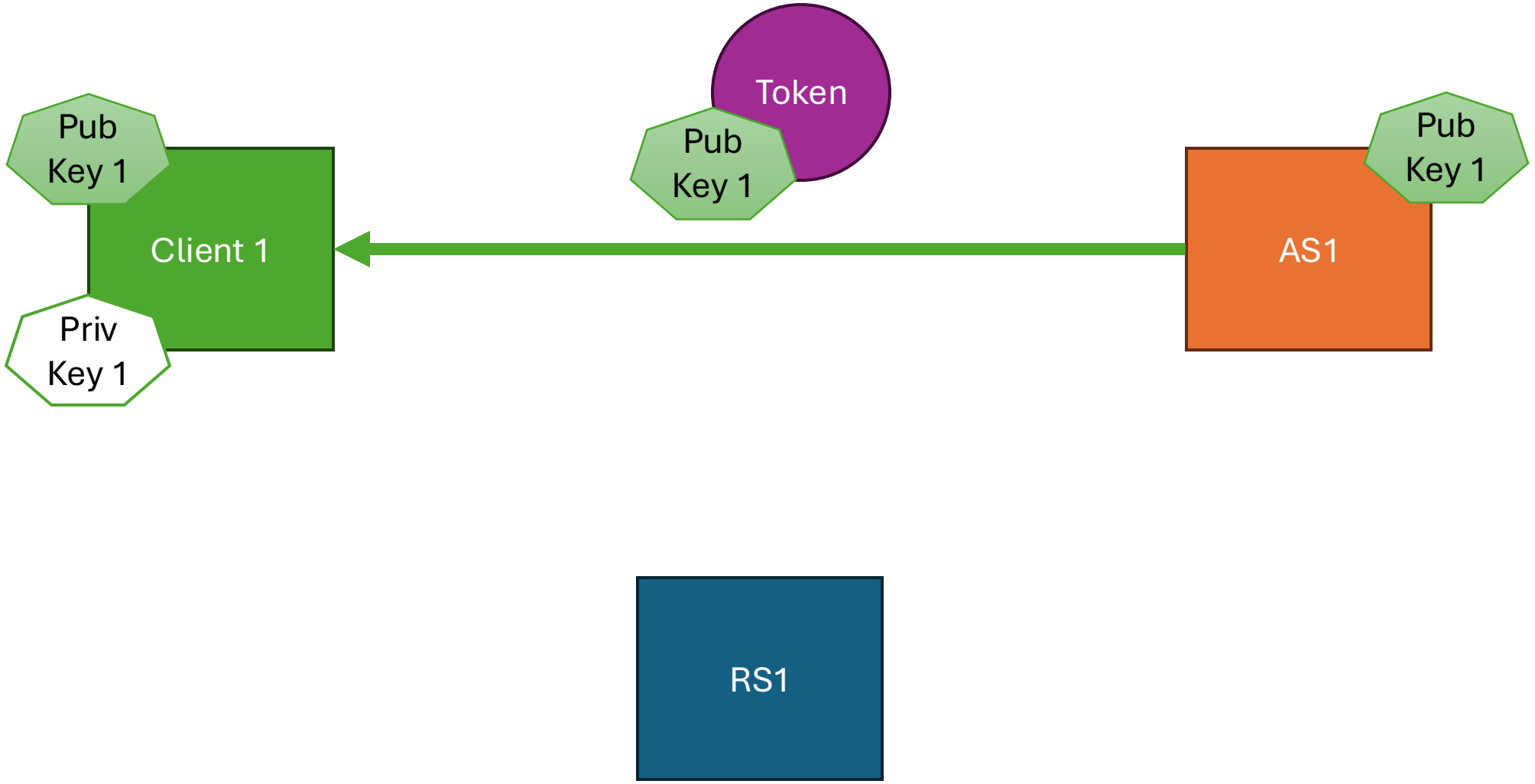
This one is secret

Need access to the secret to create this proof

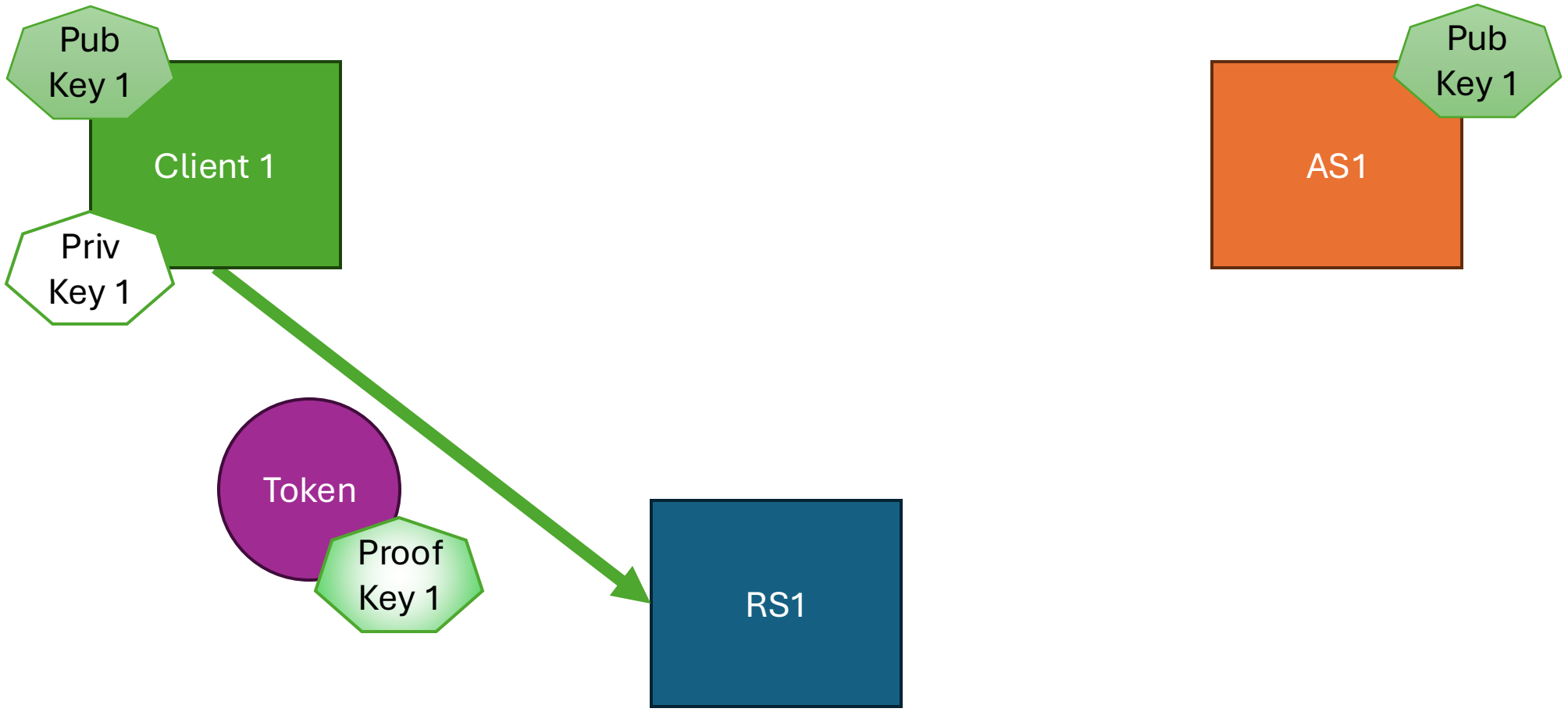
Present proof of possession of key 1 to request a bound token



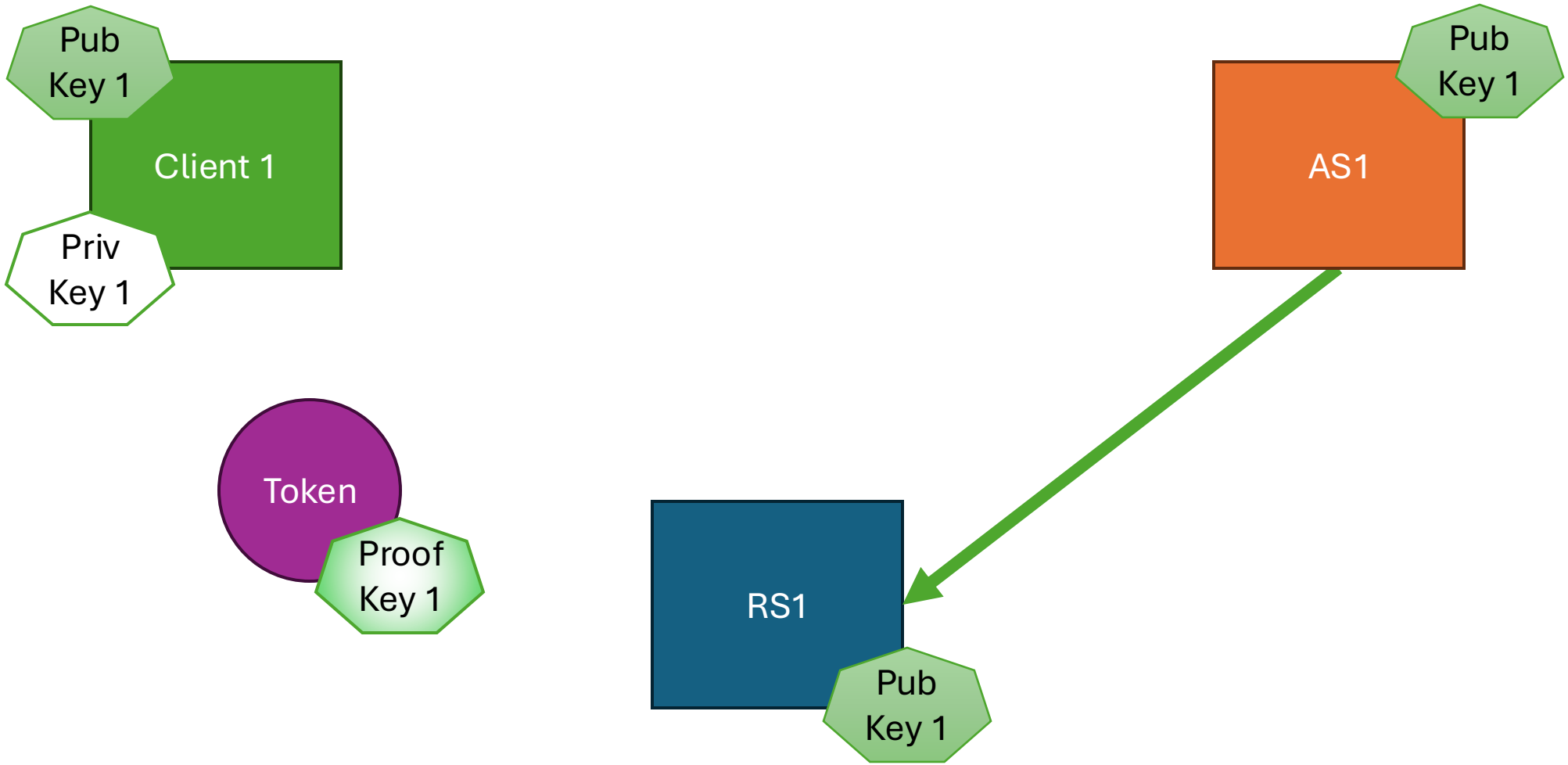
Get a token bound to key 1



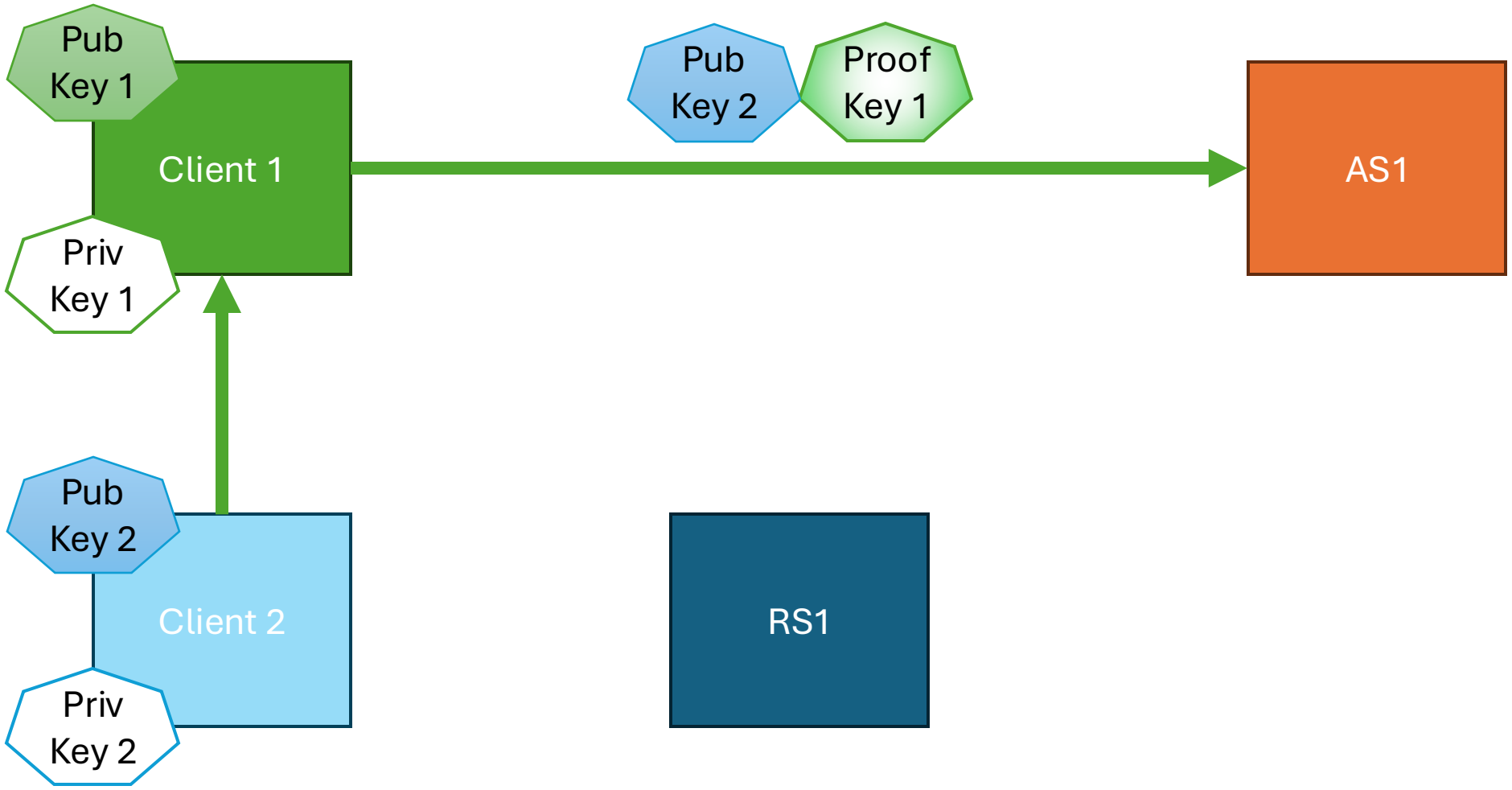
Present a token with a proof of key 1



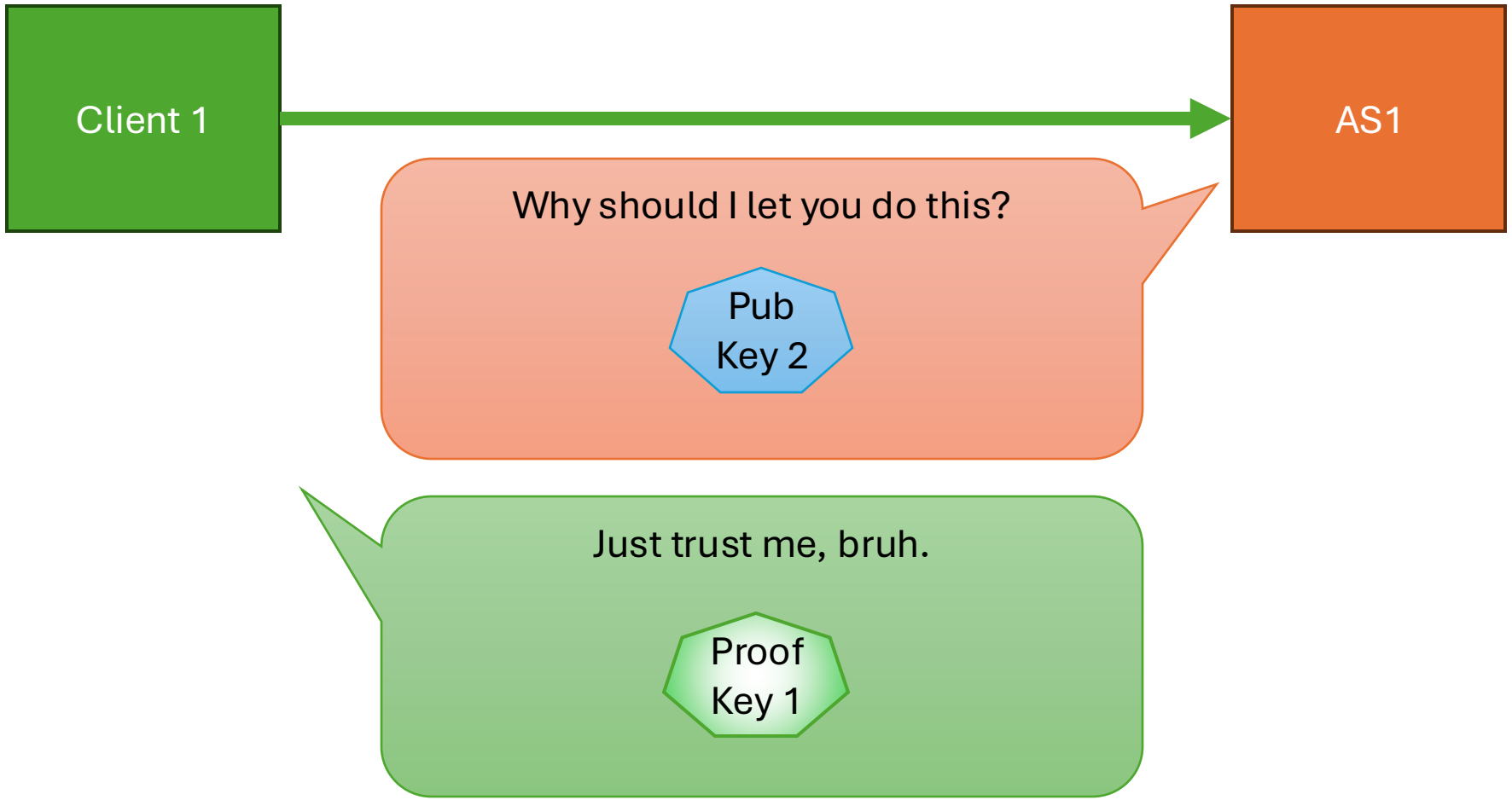
Validate proof with access to public key



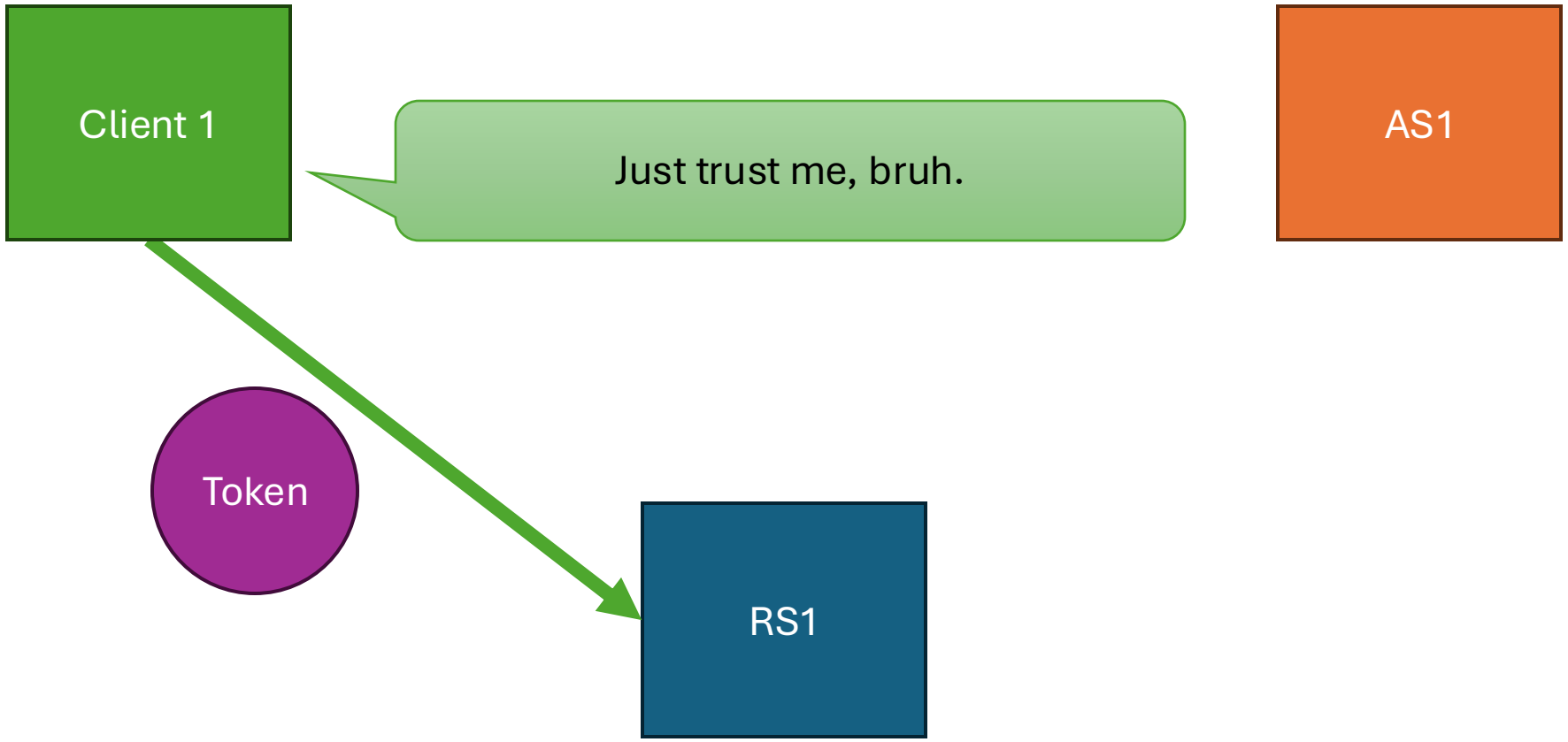
But if I want to get a token bound to a key I don't have



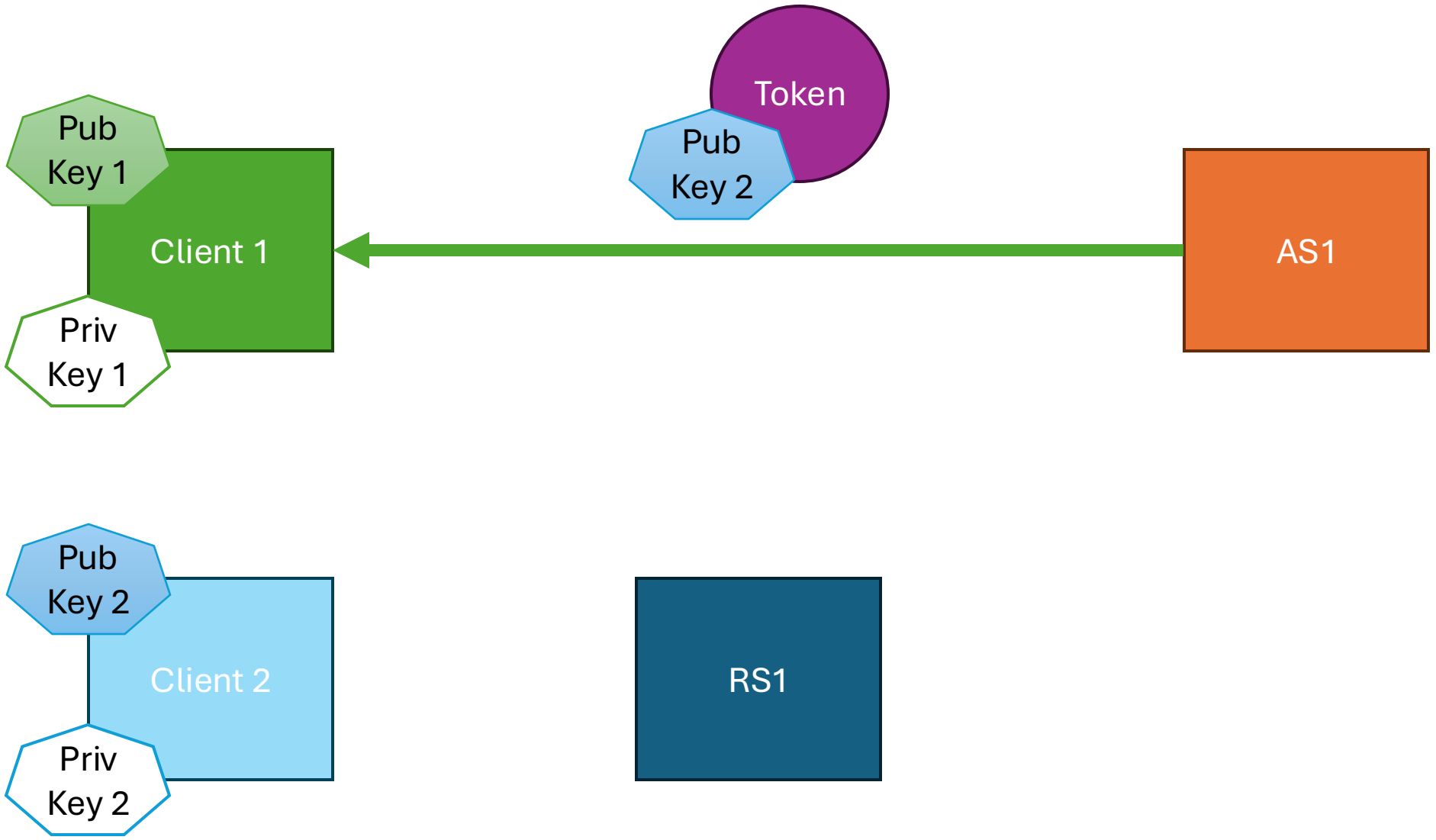
The proof doesn't line up with the request



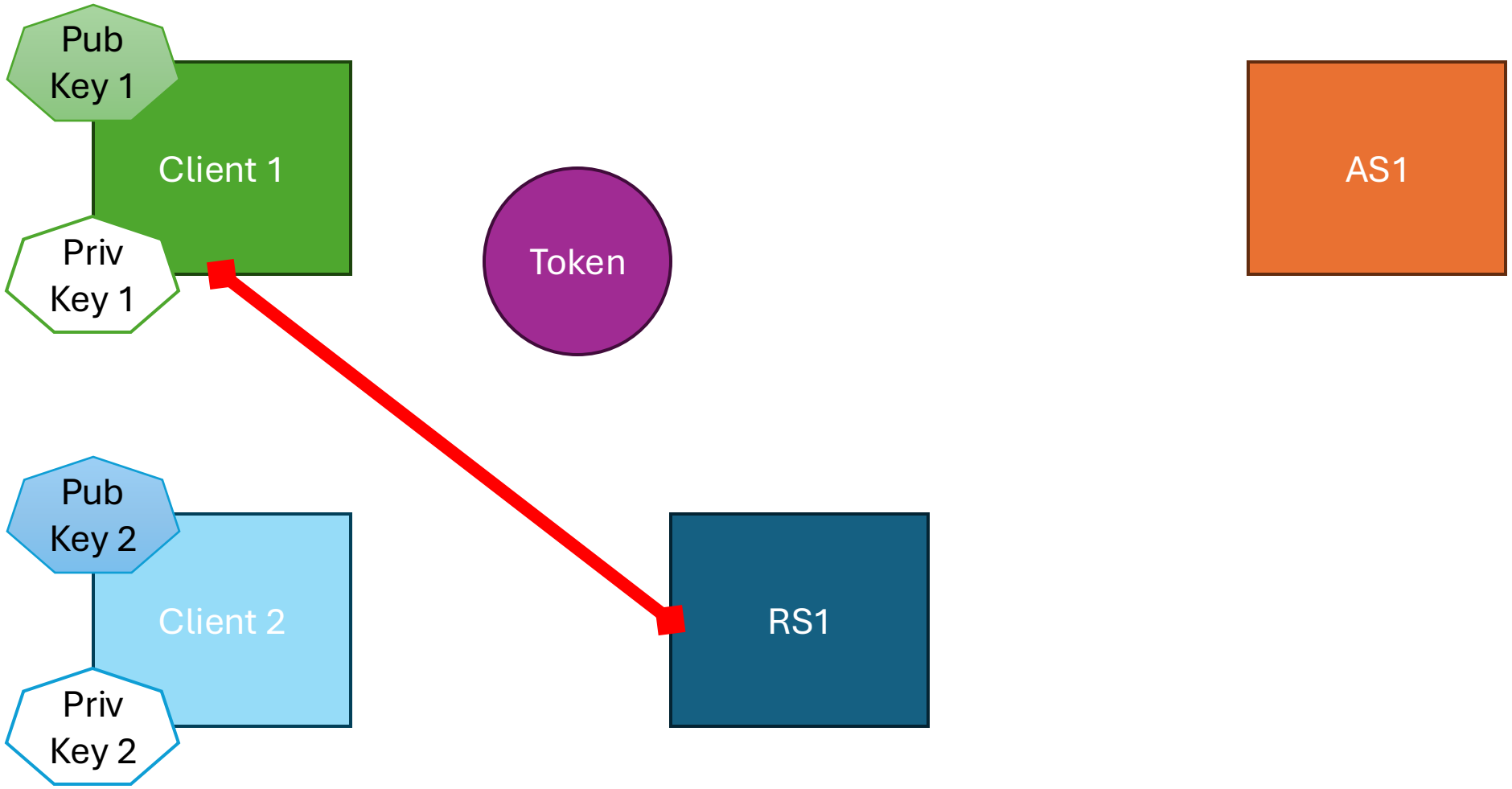
Bearer tokens already kinda do this, it's keys that make it weird



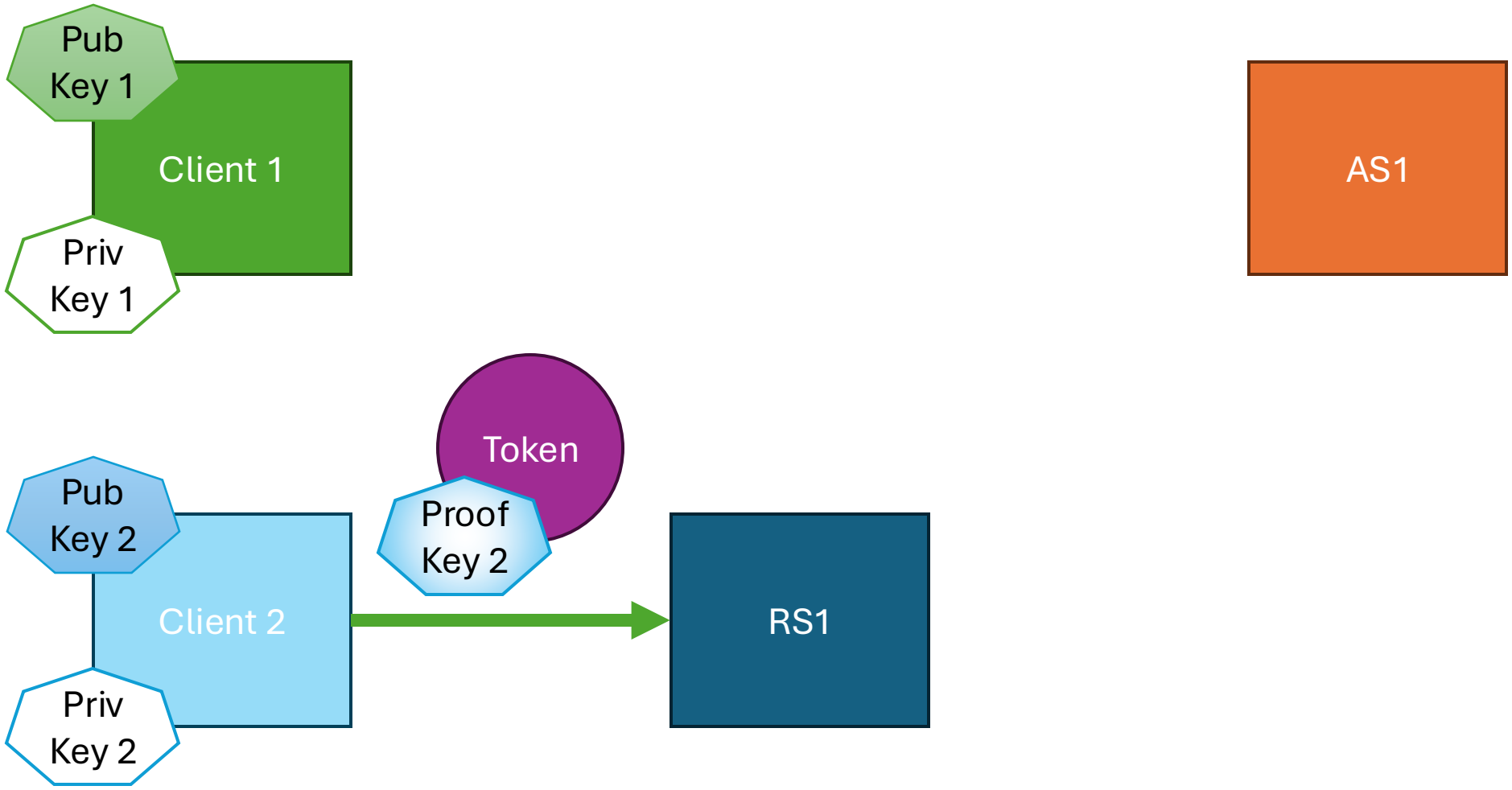
The desired result



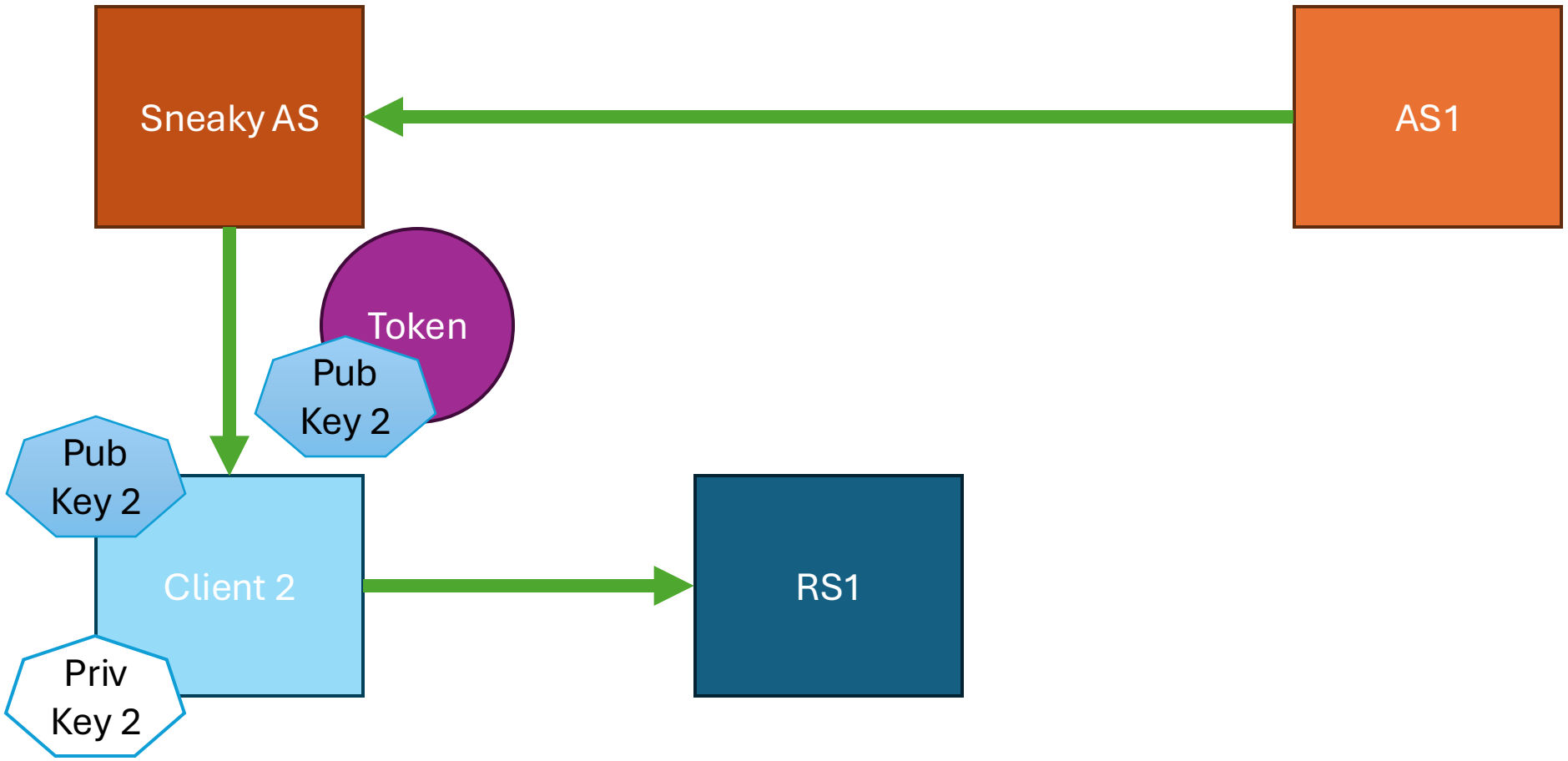
This does not work (by design)



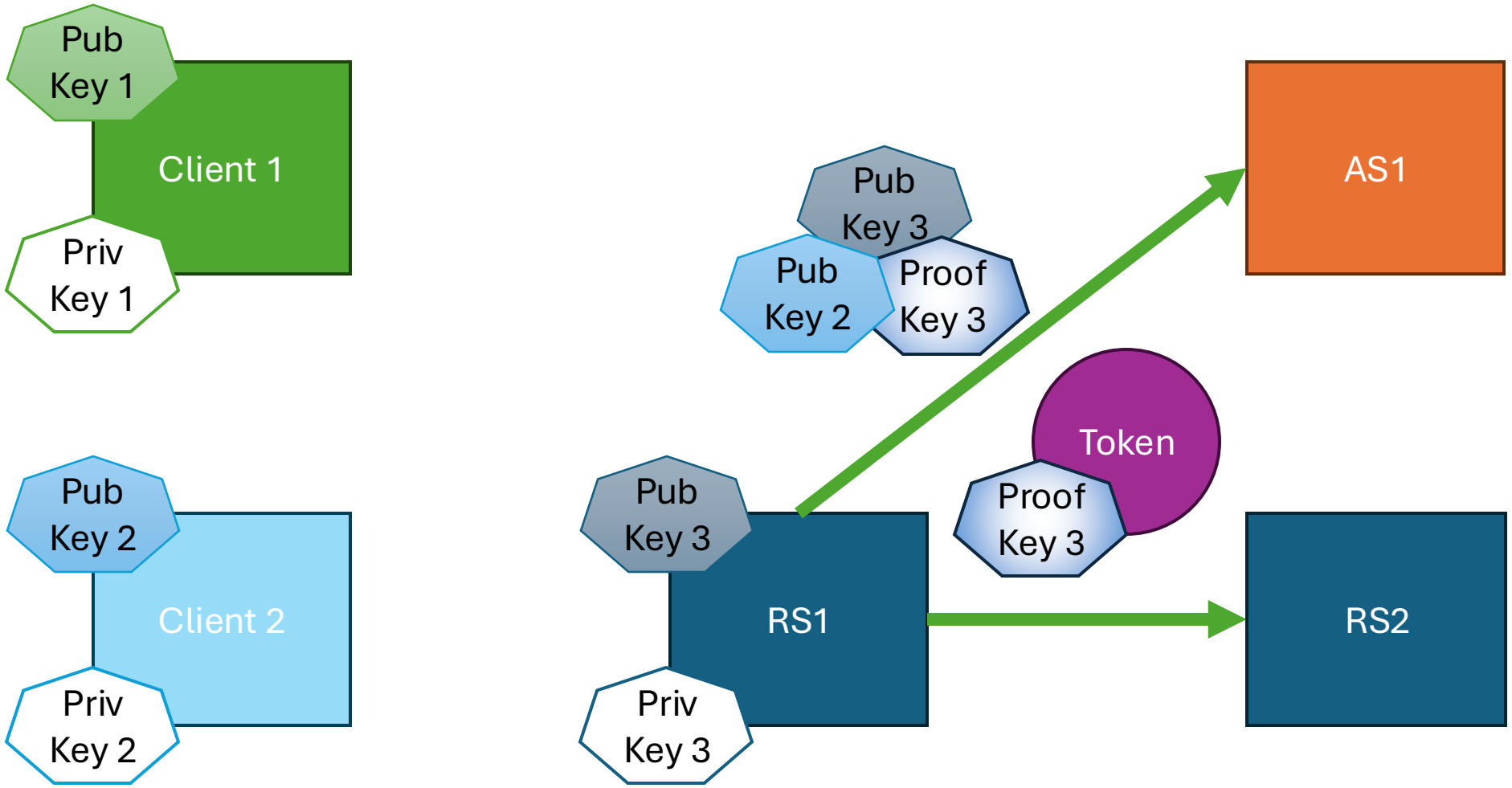
This does work



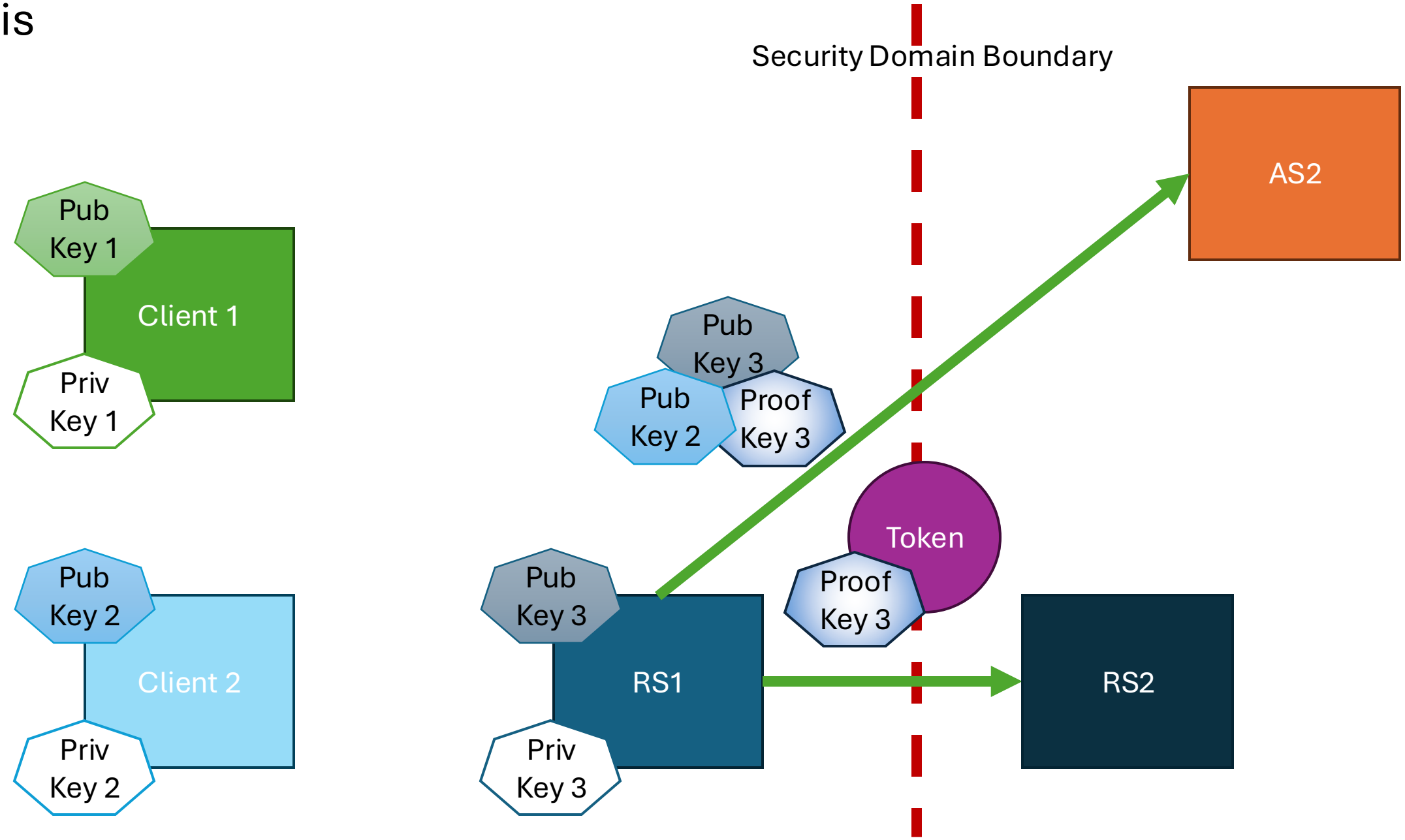
Why could this maybe be bad?



And then there's this



And this


Security Domain Boundary
AS2
Pub
Key 1
Client 1
Priv
Key 1
Pub
Key 3
Pub
Key 2
Proof
Key 3
Token
Proof
Key 3
Pub
Key 2
Client 2
Priv
Key 2
Pub
Key 3
RS1
RS2
Priv
Key 3

So what do we do about it?

- It keeps coming up in conversations and drafts
- So far we're pretty hand-wavy about it
- It's a different set of security assumptions and properties than what OAuth claims to be built on
- Different patterns for getting what we're after
- Do we define a pattern? Define protocol parameters?

We could at least document it

https://datatracker.ietf.org/doc/html/draft-richer-oauth-tmb-claim

The draft is tongue-in-cheek but the problem is not