Key Negotiation Protocol
&
Trust Router

draft-howlett-radsec-knp

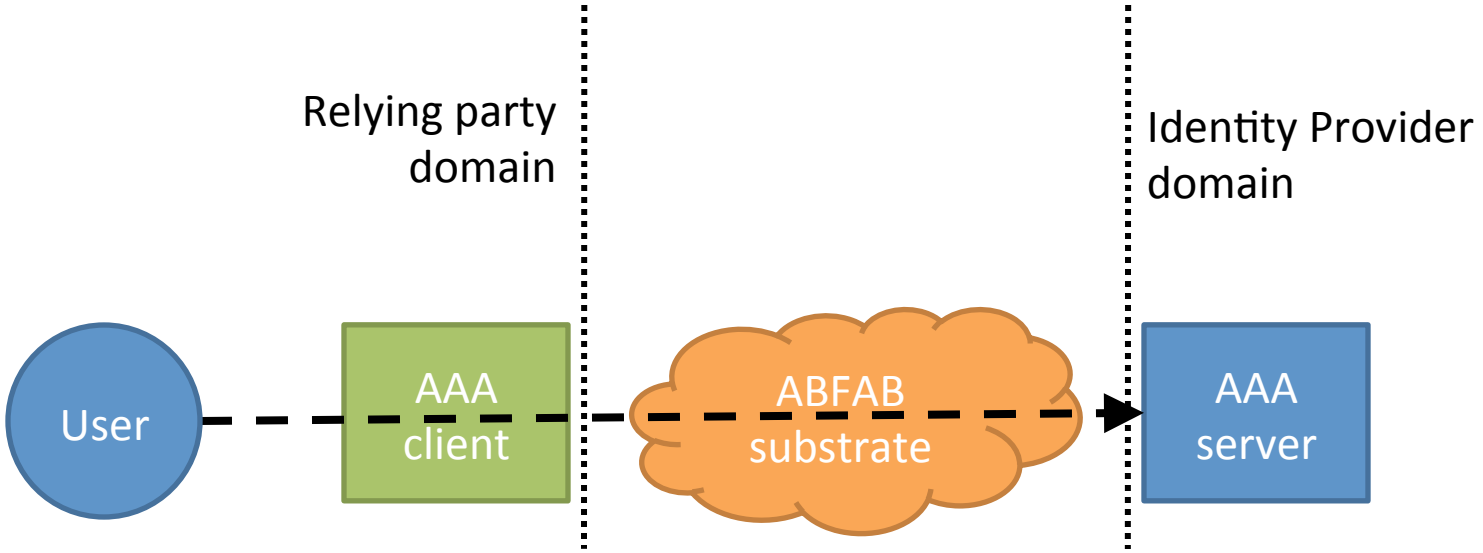
ABFAB, IETF 80

31 March, Prague.

Introduction

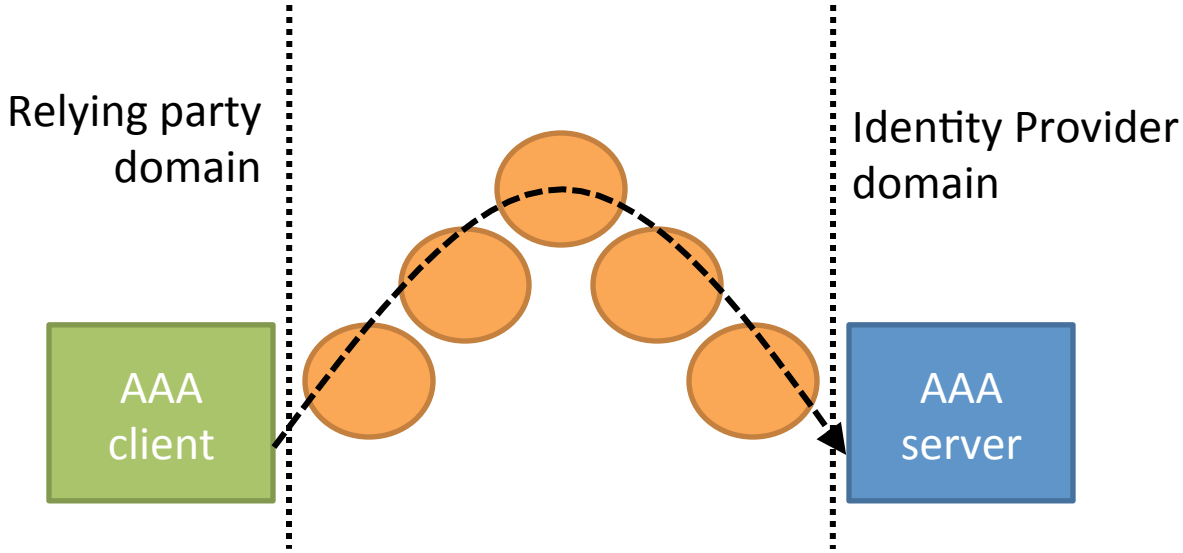
- The ABFAB architecture does not require any particular AAA strategy for connecting RPs to IdPs.
- This presentation describes a particular strategy that has some advantages over some existing strategies.
- The good news: the technology is very simple.
- The bad news: the motivations are less obvious.
- Most of this presentation is about describing the problem.

ABFAB architecture



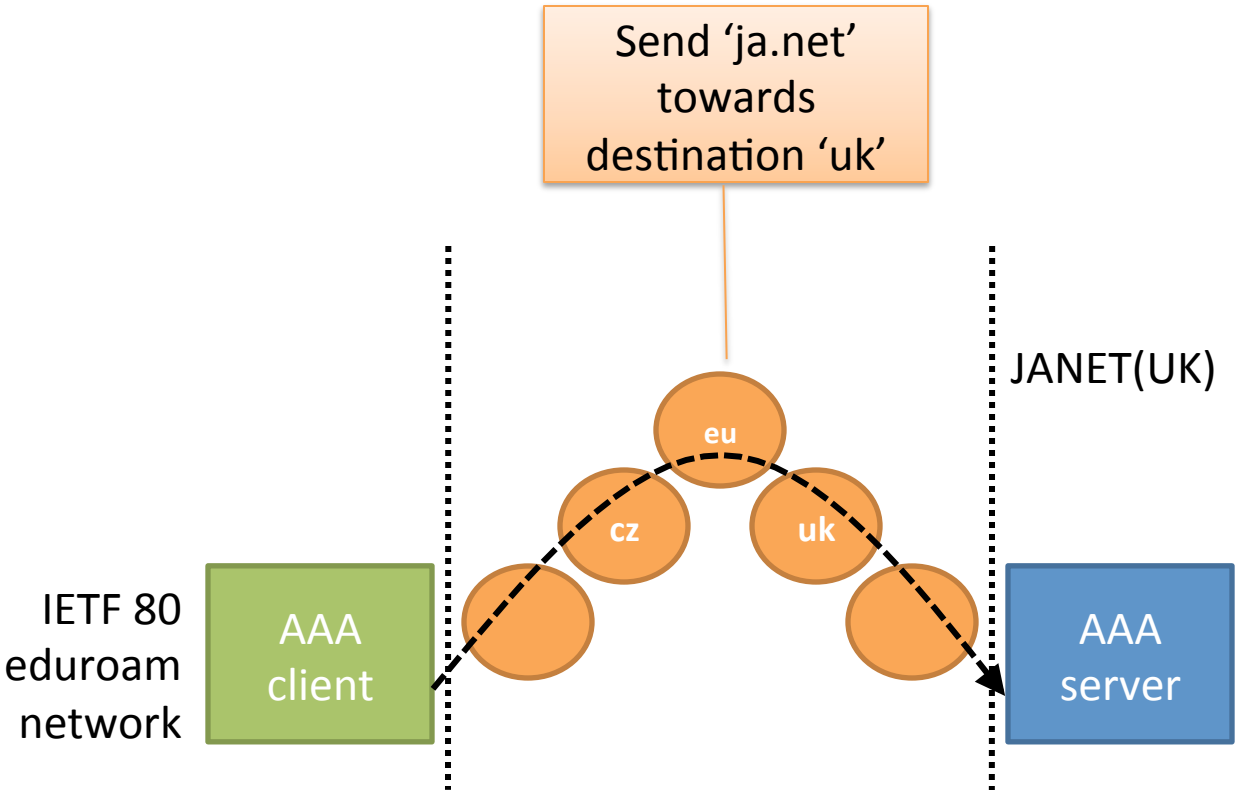
- The **ABFAB substrate** provides four functions:
  - **Transport**: how messages are conveyed between client and server
  - **Server discovery**: how messages find a server
  - **Trust establishment**: how the client/server establish confidence that they are talking to the right client/server.
  - **Rules determination**: how the client/server decide what they should infer from the messages, and how they should behave in that regime.

RADIUS substrate (1)

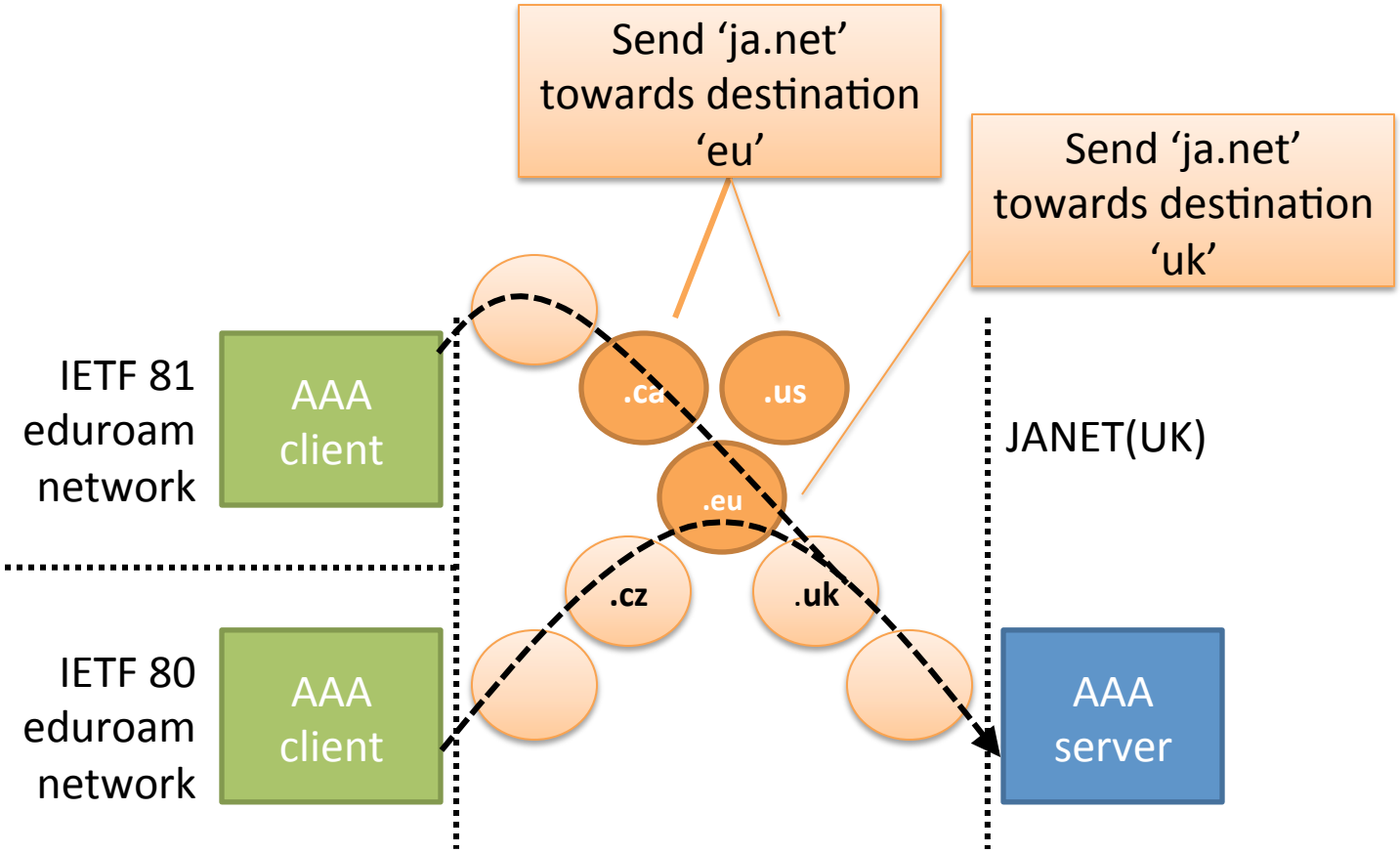


- Transport — Hop-by-hop UDP datagram
- Server discovery — Hop-by-hop realm matching, **static configuration at each hop**.
- Trust establishment — Hop-by-hop shared secret, **static configuration at each hop**.
- Rules determination — Locally configured policy, **static configuration at each hop**.

Static configuration is simple…



…until it isn't.



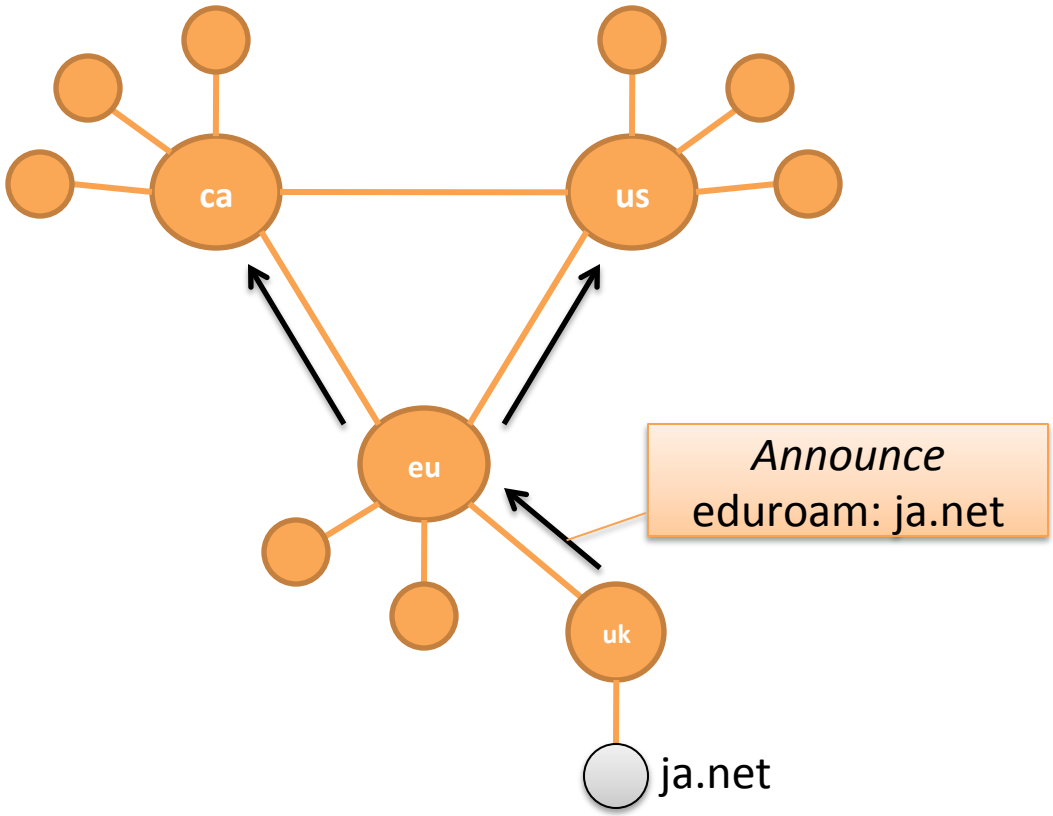
Static configuration doesn't scale…

- As an AAA system scales, you need to maintain more configuration across more nodes.
- The configuration is necessarily dissimilar between AAA nodes, but the entire system needs to behave as though all nodes share a consistent view of the entire system. Inconsistency may result in undesirable behaviour.
- Inventing an *ad hoc* solution within a single domain is trivial. The multi-domain case is also tractable, providing there is close coordination.
- However, if ABFAB is successful the potential number of domains and overall system size is considerable: coordination will be challenging.
- We need a standard mechanism that enables AAA nodes within a large and loosely-coupled AAA system to behave as though they share a consistent view of the entire system.

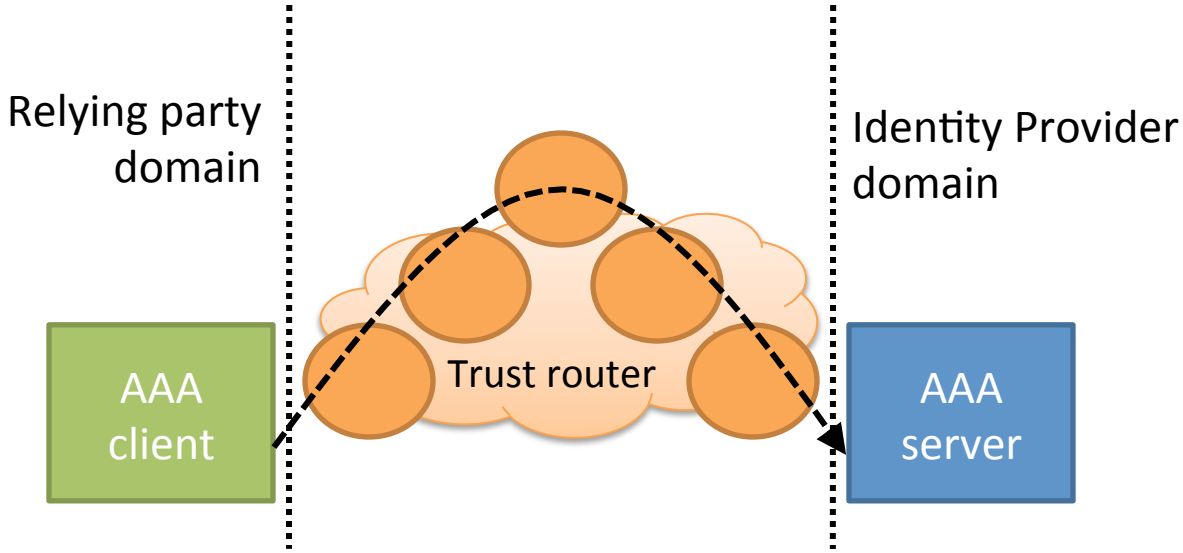
…that's why we have routing protocols

- We already have a protocol that allows IP routers to replicate routing configuration: BGP.
- What if AAA configuration could be replicated between AAA nodes using a 'trust router' protocol?
- AAA nodes could use this protocol to advertise:
  - NAI realms: for server discovery.
  - Rules regimes: for rules determination.

Trust router protocol



RADIUS substrate (2)



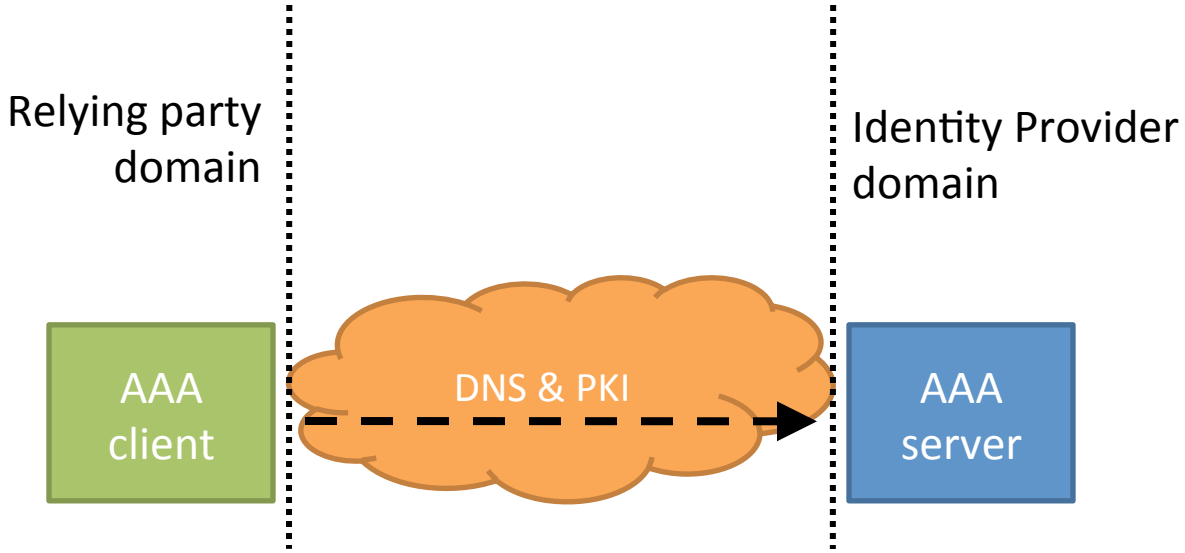
- Transport — Hop-by-hop UDP datagram
- Server discovery — Realm matching using Trust Router protocol
- Trust establishment — Hop-by-hop shared secret**, static configuration at each hop**
- Rules determination — Trust router protocol; peer known implicitly.

Well, we have RadSec…

- RadSec is RADIUS over TLS or DTLS
- Invoke PKI to banish hop-by-hop security; permits e2e trust establishment.
- Knowing your peer explicitly may improve rules determination.

- Other benefits:
  - Prevents exposure of information to intermediate AAA nodes.
  - Reduces EAP transmission latency.

RadSec substrate (1)



- Transport — TLS/TCP
- Server discovery — DNS
- Trust establishment — PKI
- Rules determination — Locally configured policy, peer known explicitly; **static configuration at each hop**.

A single PKI for ABFAB deployments?

- A PKI environment is a one-to-many relationship; an issuer's policies may impose costs on some subset of those RPs that are not relevant to their business relationship(s).
- A one-to-one relationship allows the actors to agree their requirements without consideration of irrelevant actors in the system.
- But pairwise credentials don't scale, right?

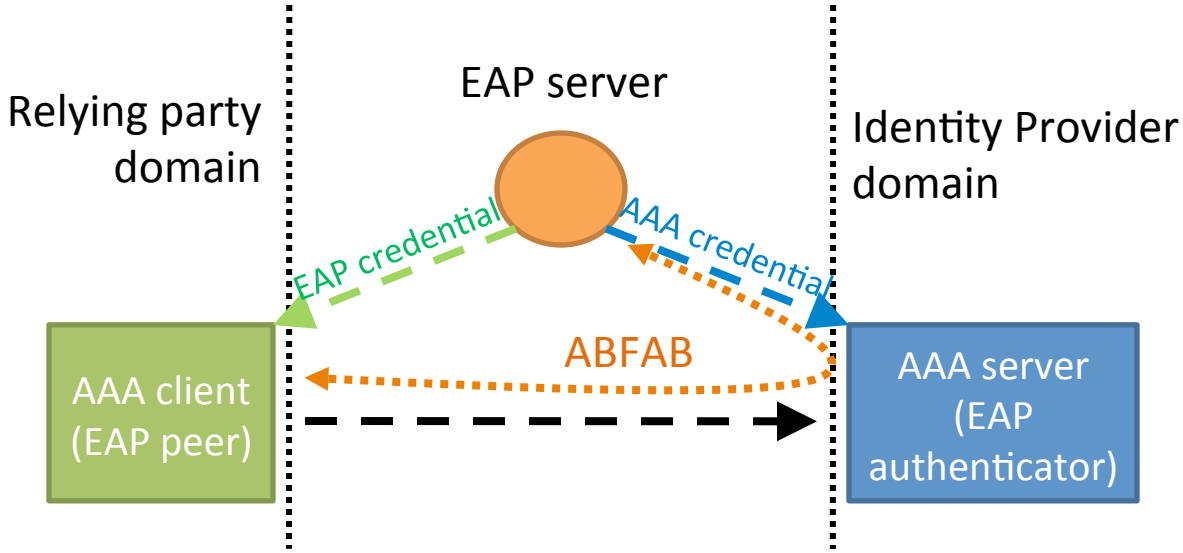
Didn't we just fix the multiple credential problem?

- We've just invented a mechanism that enables a single EAP credential to be used against all RPs that trust the EAP server.

- An AAA server is just another RP; let's apply ABFAB to RadSec!

- "WTF!" is a perfectly understandable response at this point.

RadSec substrate (2)

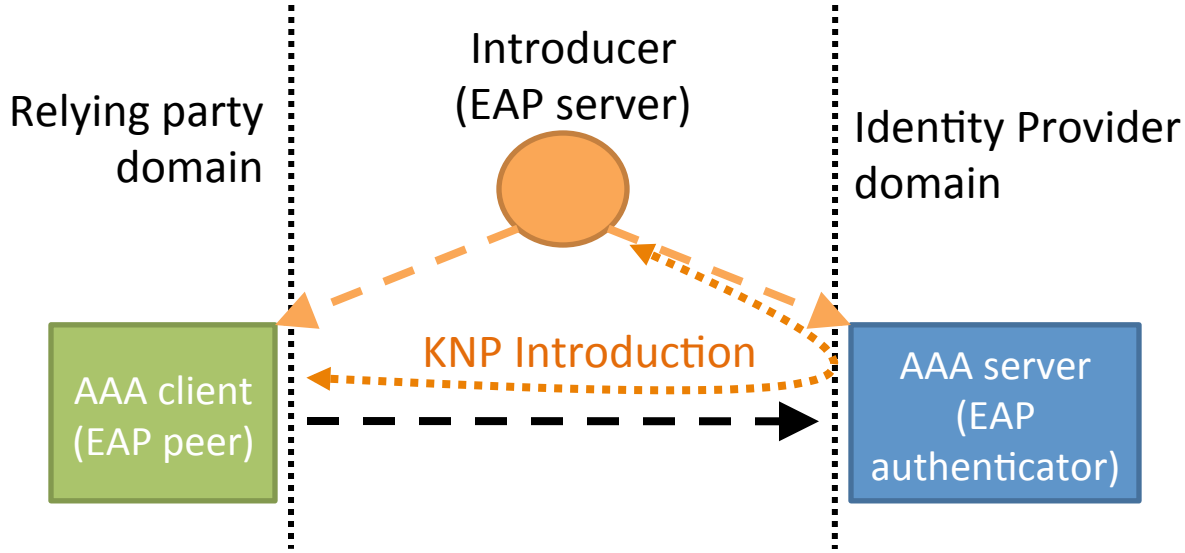


- Transport **TLS/TCP**
- Server discovery **Trust Router**
- Trust establishment **ABFAB**
- Rules determination **Trust Router**

Key Negotiation Protocol

- KNP enables a RadSec client and server to dynamically establish a short-lived credential for a subsequent RadSec connection.
- KNP uses EAP authentication of credentials issued to the AAA client by an EAP server that is also trusted by the AAA server.
- The EAP server is called the ‘Introducer’. The process of establishing the RadSec credential between AAA client and server is called ‘Introduction’.

KNP substrate

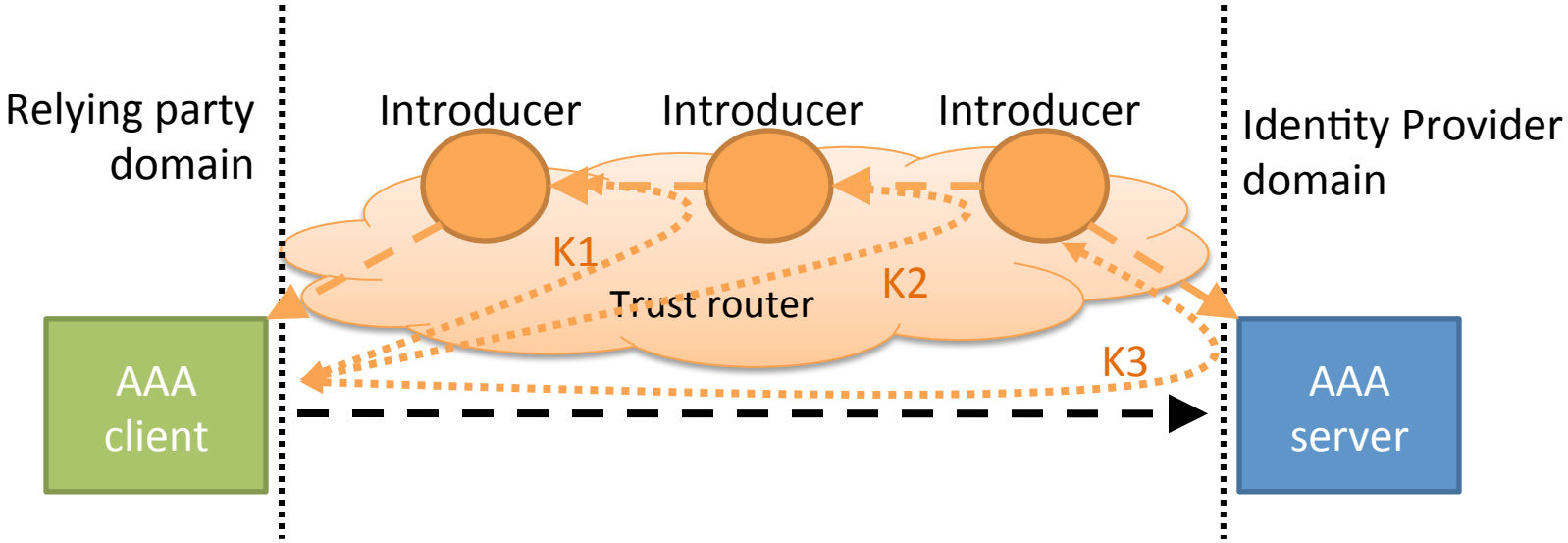


- Transport **TLS/TCP**
- Server discovery **Trust Router**
- Trust establishment **KNP Introduction**
- Rules determination **Trust Router**

Transitive operation

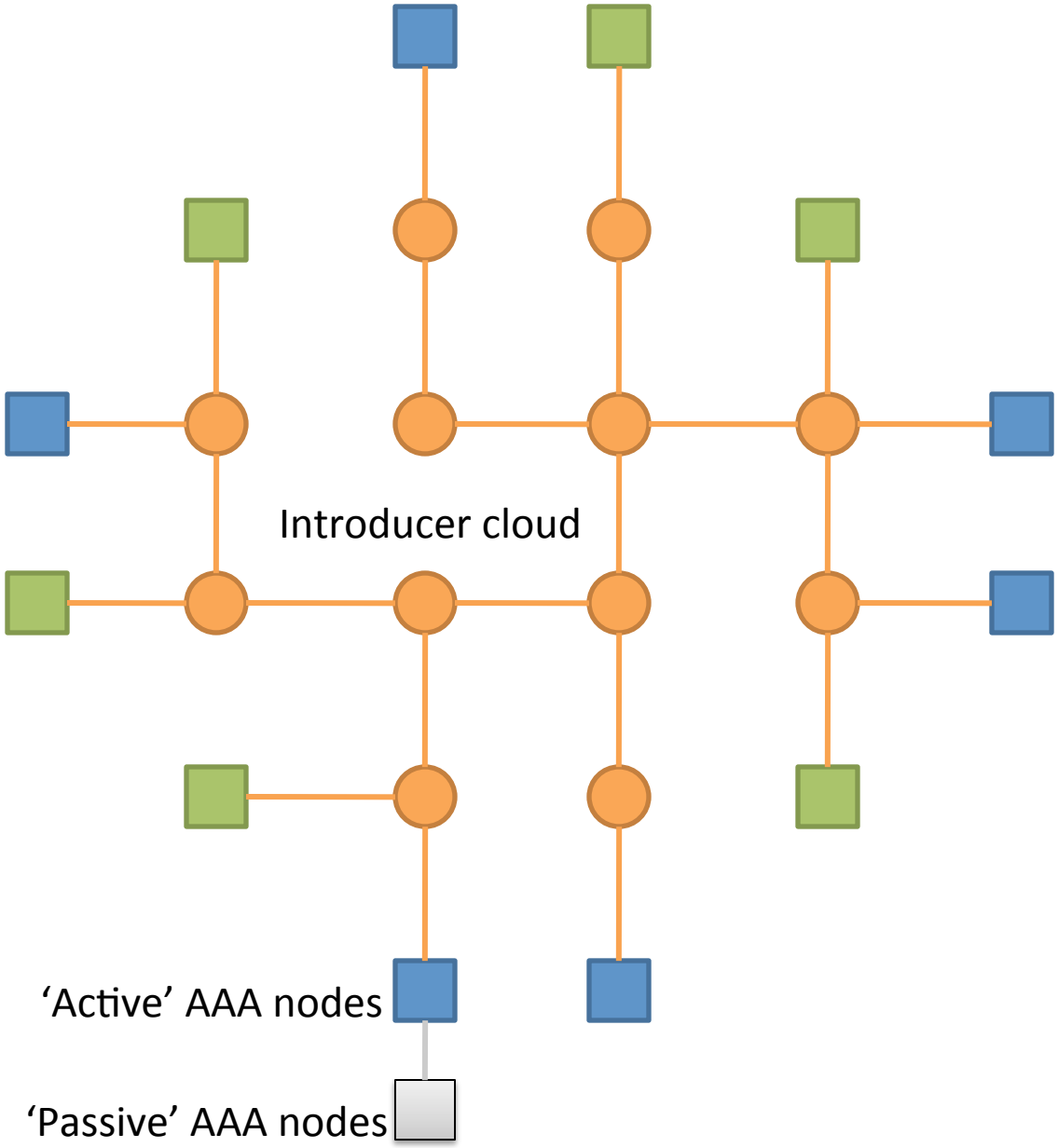
- Not all AAA nodes share a common Introducer.
- An Introducer can also be party as AAA client or server to an Introduction.
- This enables transitive introduction: the AAA client recurses along a path of Introducers to the AAA server.

Transitive KNP substrate



- Transport **TLS/TCP**
- Server discovery **Trust Router**
- Trust establishment **Transitive KNP Introduction**
- Rules determination **Trust Router**

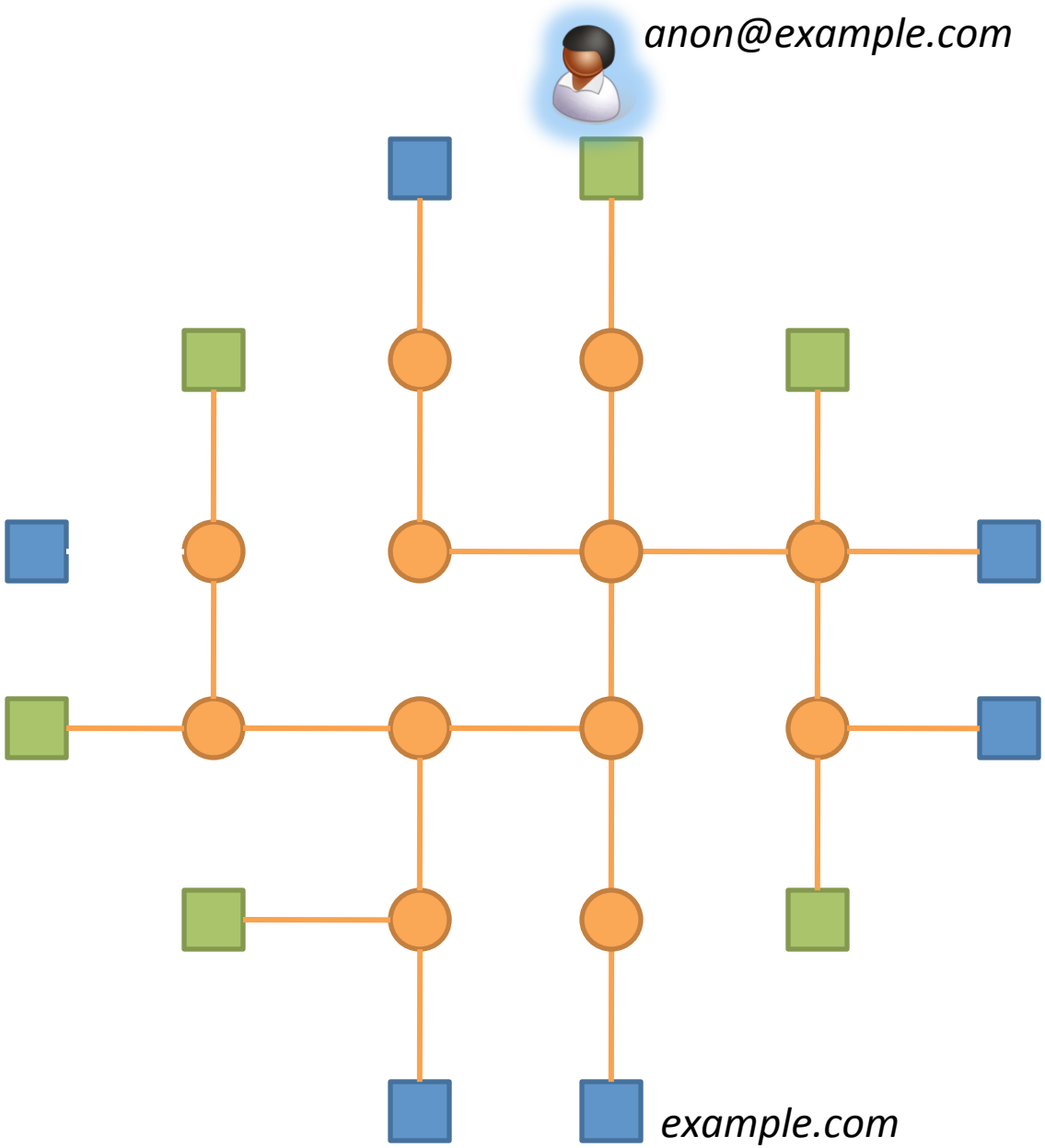
System overview



- The system actors are Introducers and KNP-aware ('active') AAA nodes.
- Introducers credential trusted AAA nodes, and each other with long-lived credentials. These probably correspond to business agreements.
- Introducers announce and consume routing configuration data (names and rules).
- Transitive KNP and these long-term credentials allow the dynamic establishment of short-lived RadSec credentials.
- The short-lived credentials may be cached to avoid repetitive recursion.
- The active nodes may be proxies for non-KNP aware ('passive') AAA nodes.

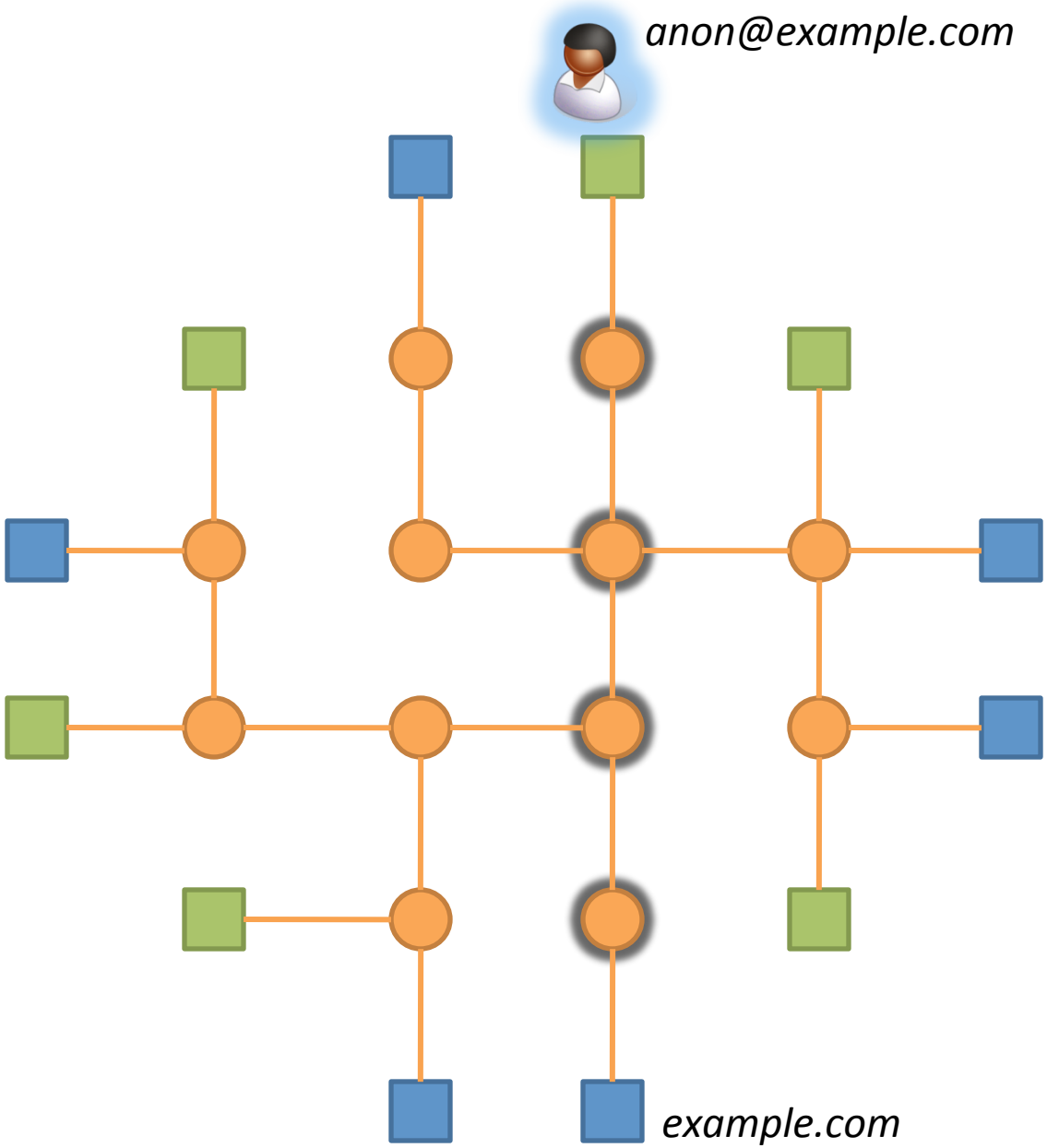
Conclusions

- RadSec KNP places the costs associated with establishing a business relationship with the parties
- ABFAB architecture by providing a substrate with properties that are particularly suitable for loosely-coupled systems.
- KNP is itself an application of ABFAB, that re-uses existing components. Therefore, it does not require substantial new invention.
- Project Moonshot is planning a KNP implementation for Q3/Q4 2011.



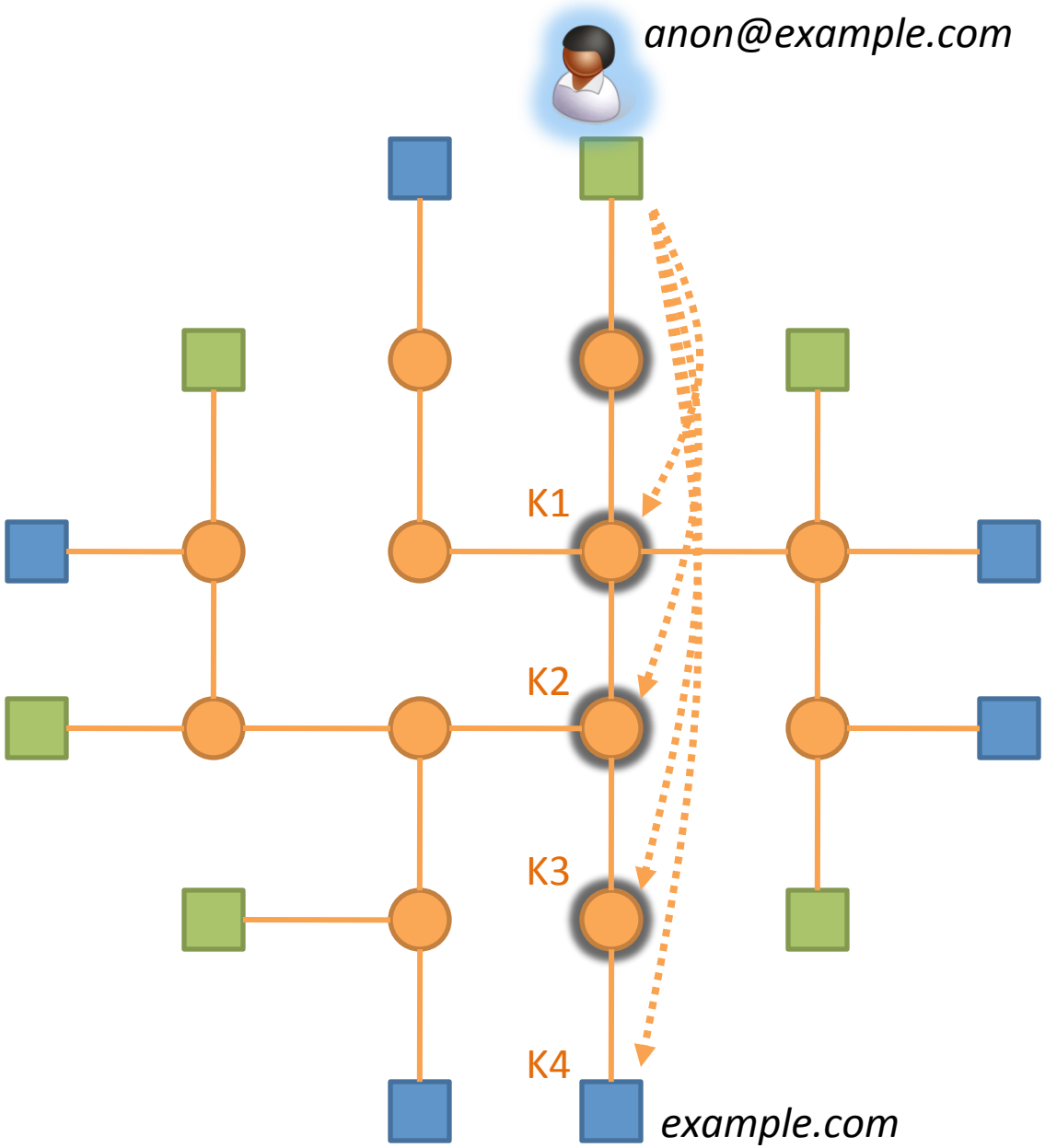
Example 1: no cached state

User connected to service. Its AAA client obtains the server realm from the user's NAI.



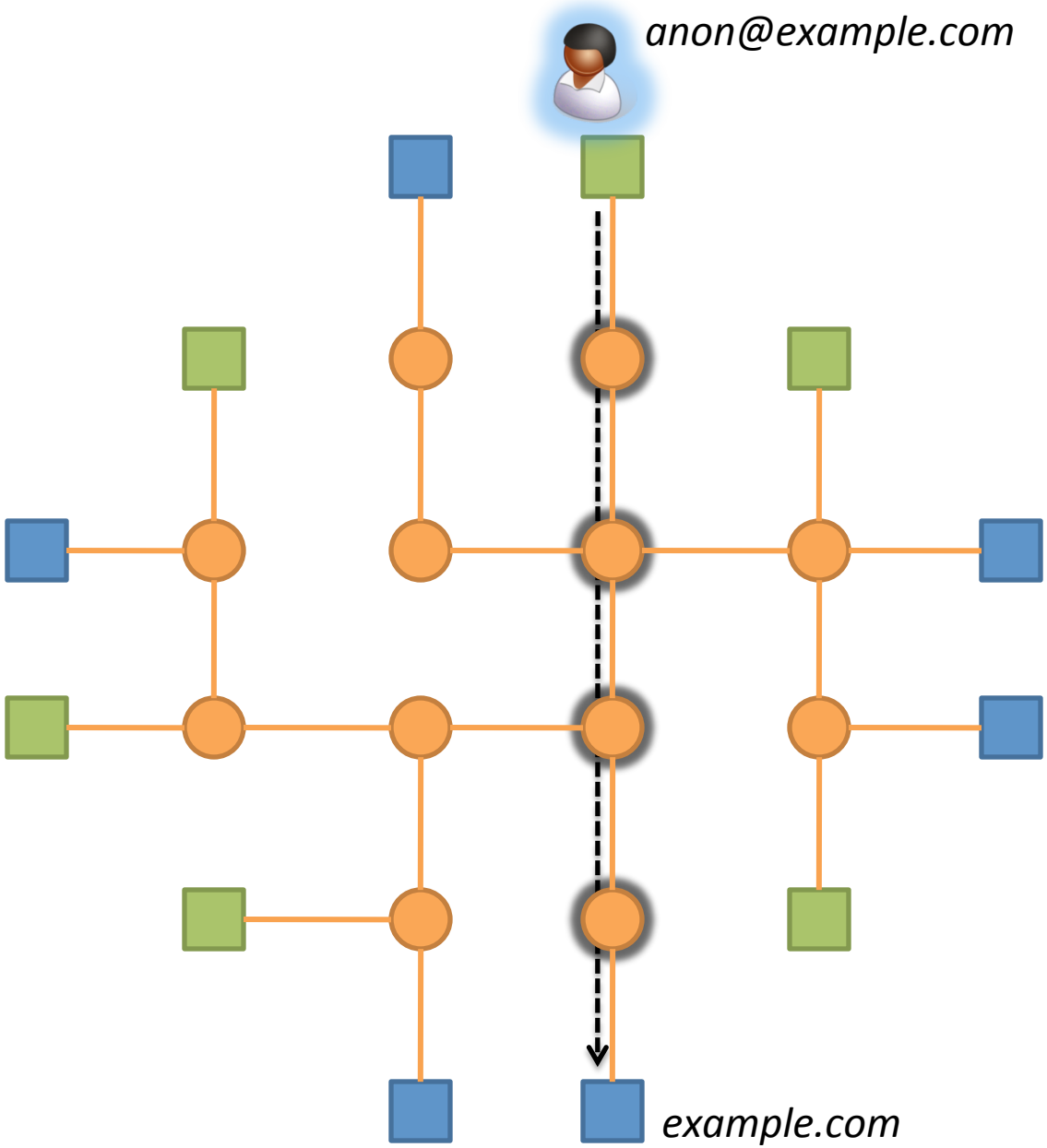
Example 1: no cached state

AAA client determines a path through the Introducer cloud to the AAA server that meets its policy.



Example 1: no cached state

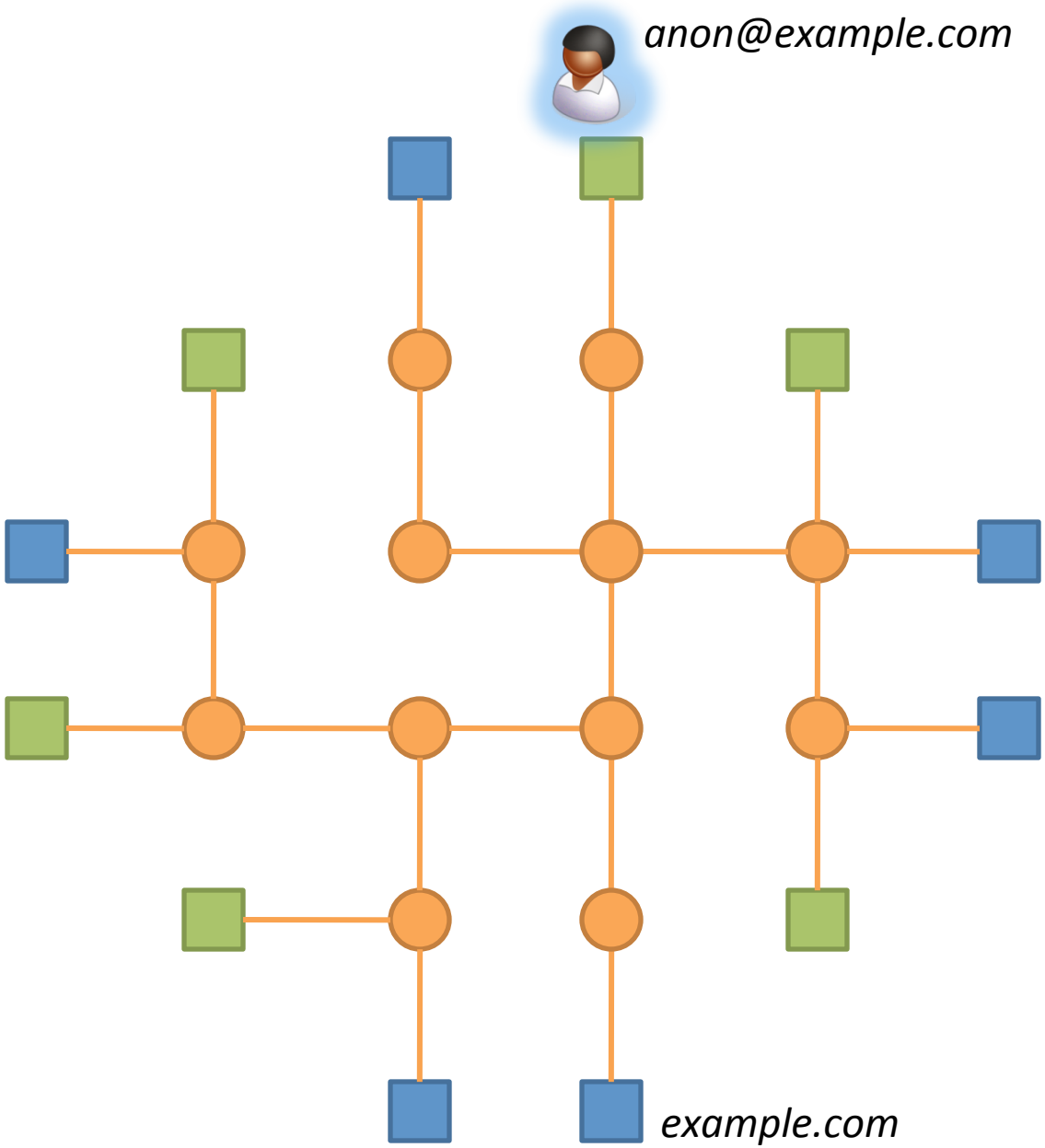
AAA client walks along the Introducer path, establishing a short-lived RadSec credential at each hop.



Example 1: no cached state

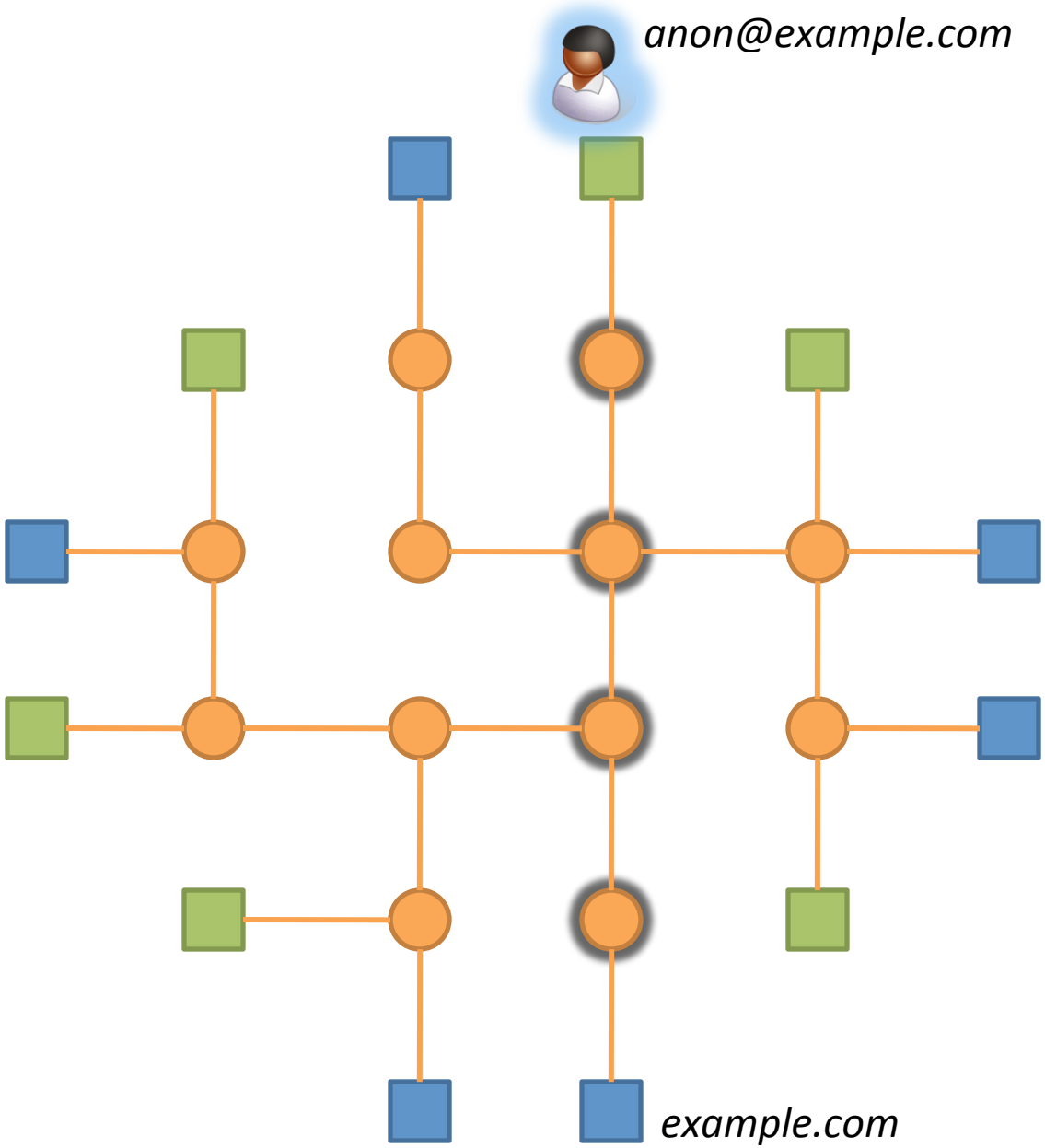
AAA client establishes a RadSec connection with the AAA server, and the user's credentials are

connection.



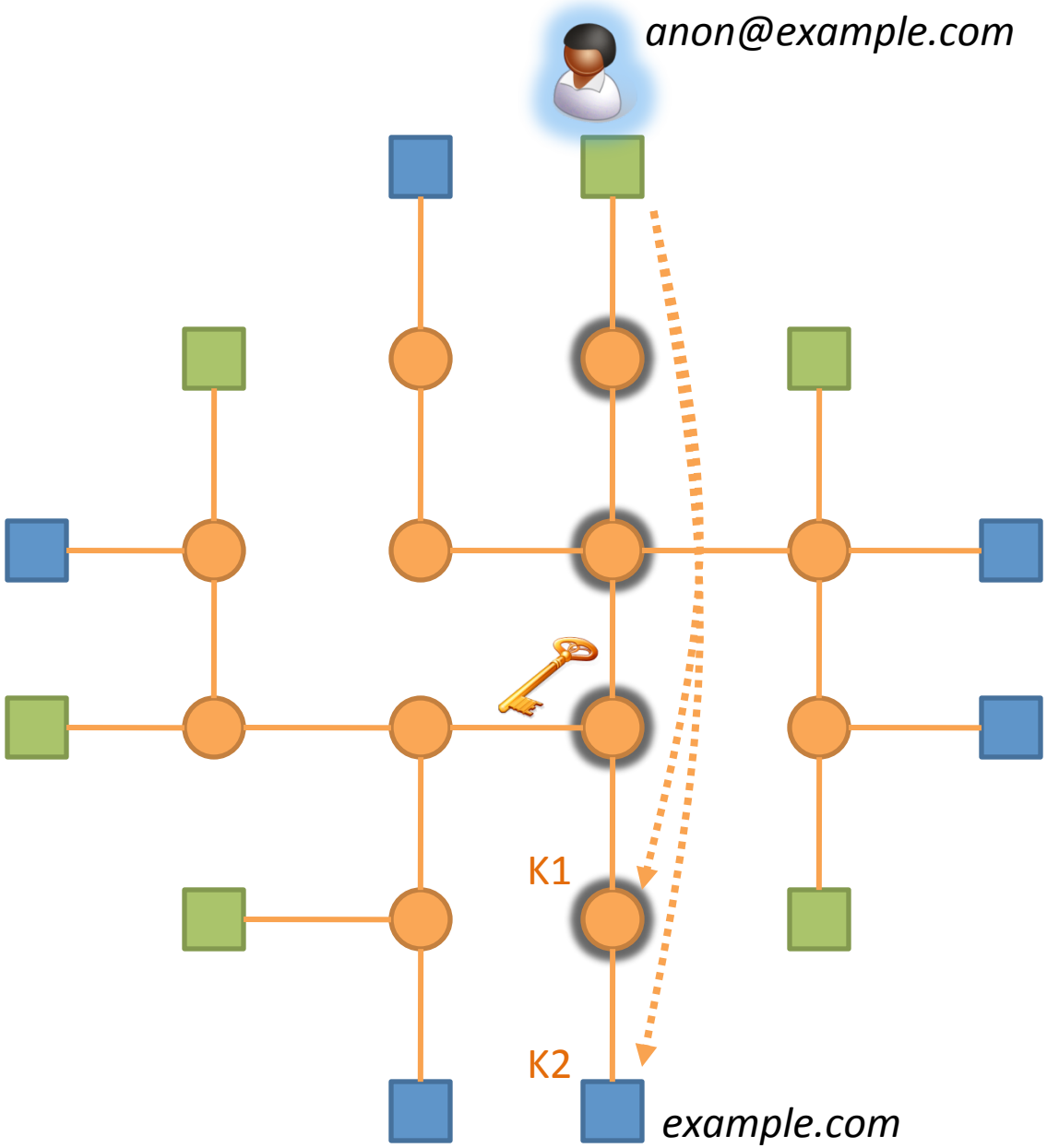
Example 2: intermediate cached state

User connects to service. Its AAA client obtains the server realm from the user's NAI.



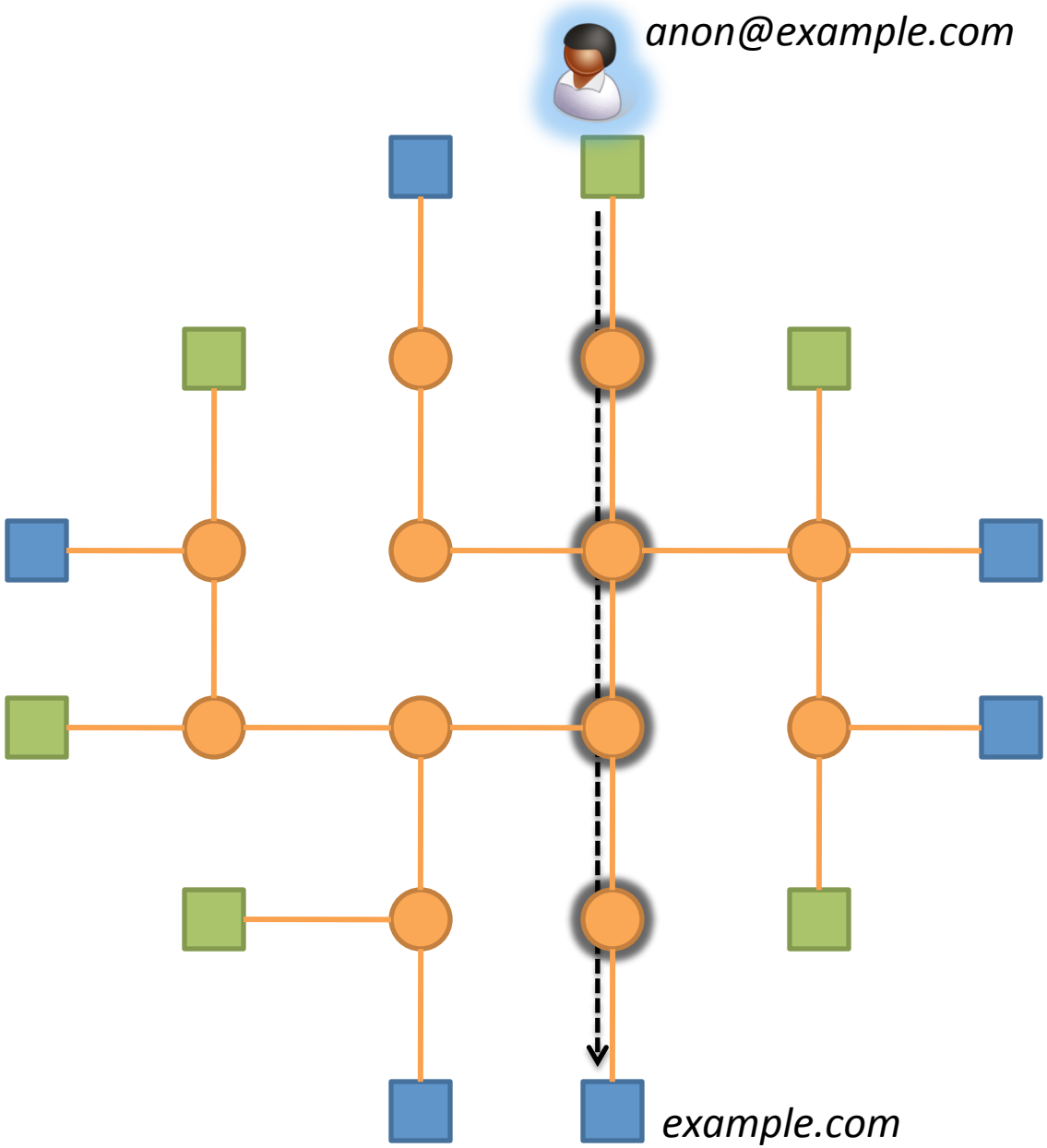
Example 2: intermediate cached state

AAA client determines a path through the Introducer cloud to the AAA server that meets its policy.



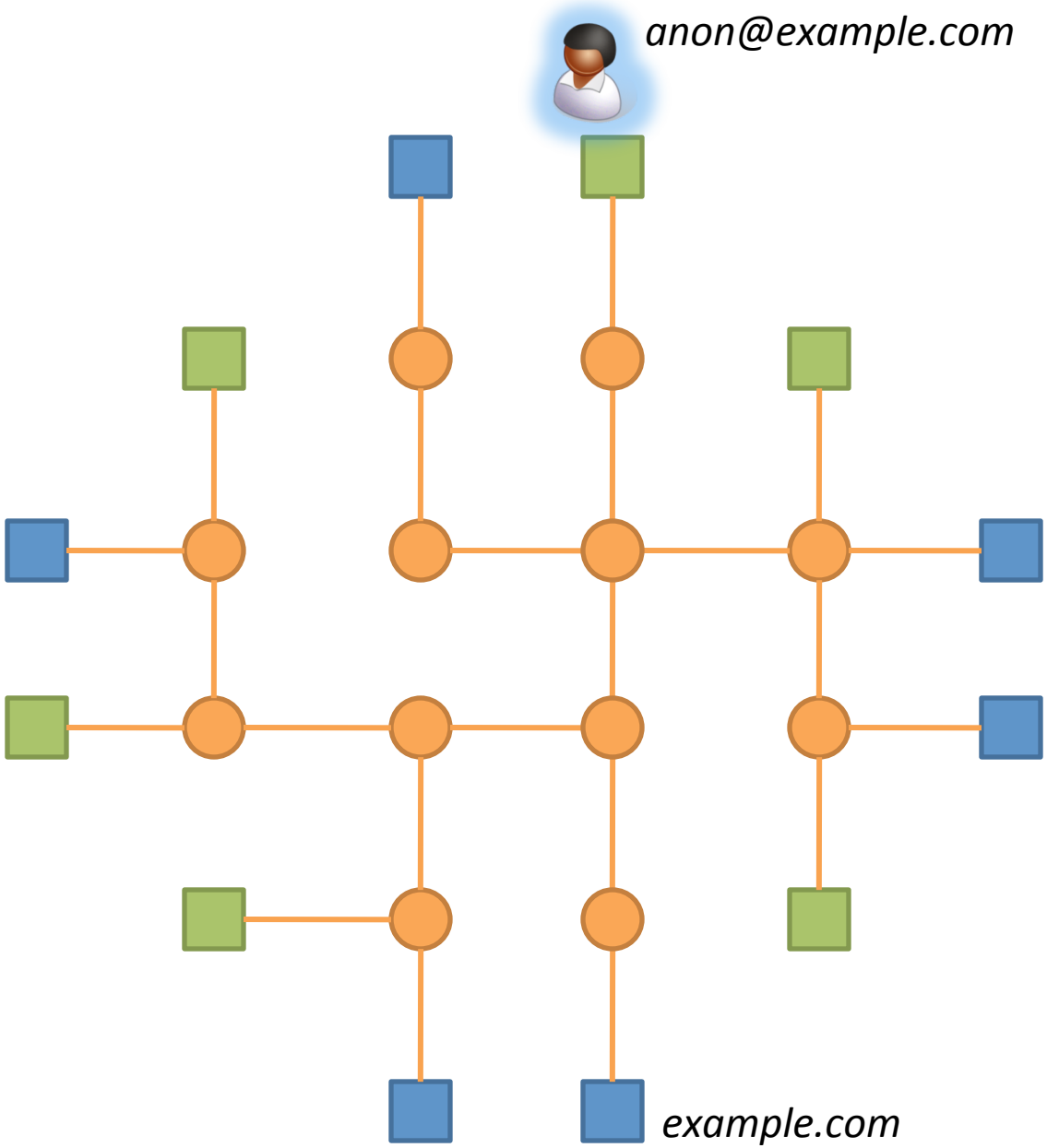
Example 2: intermediate cached state

AAA client determines that it already has a non-expired key for an intermediate Introducer. The client begins walking from this Introducer, avoiding the first two hops, establishing a short-lived RadSec credential at the subsequent hops.



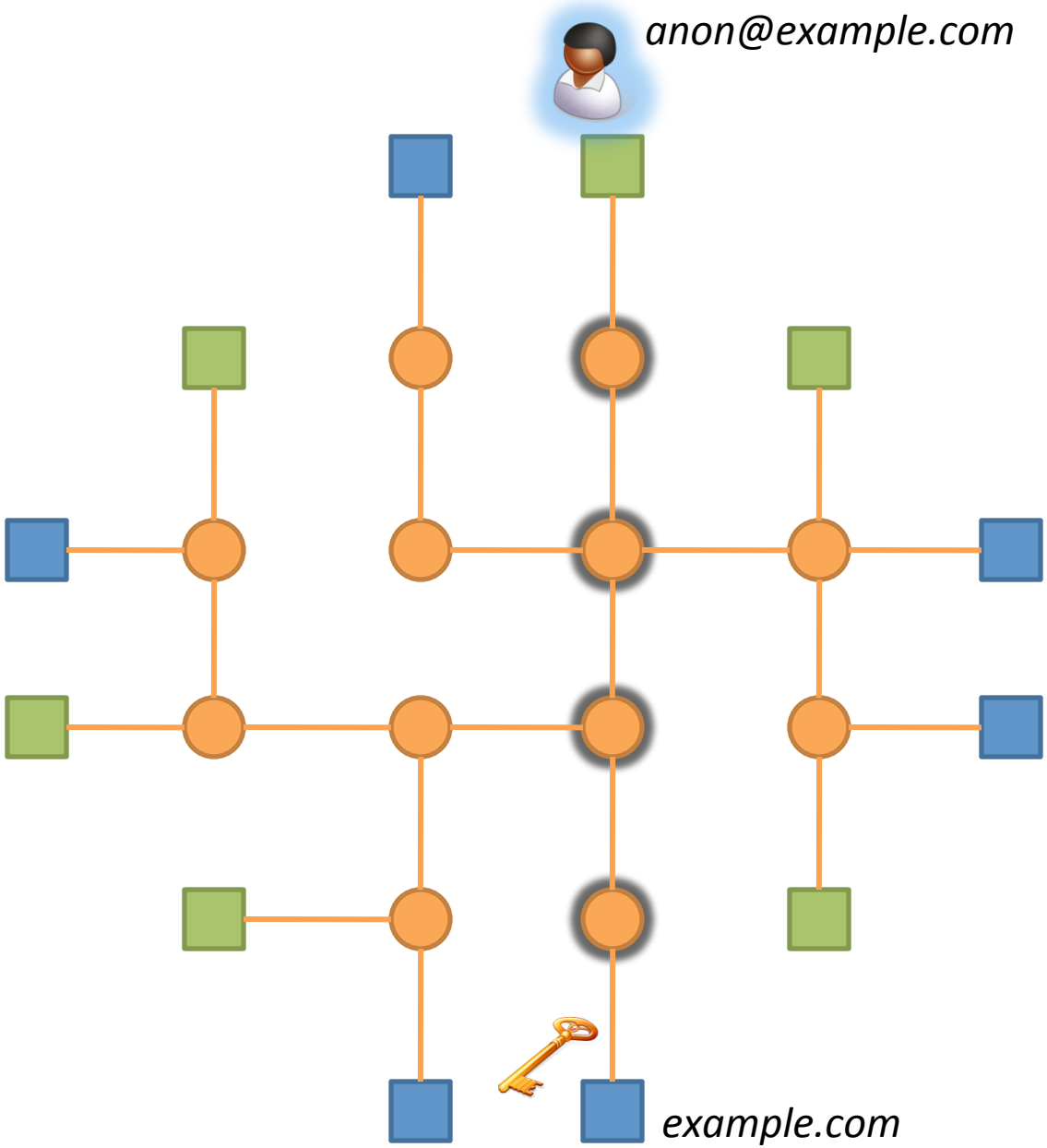
Example 2: intermediate cached state

AAA client establishes a RadSec connection with the AAA server, and the user's credentials are authenticated across this connection.



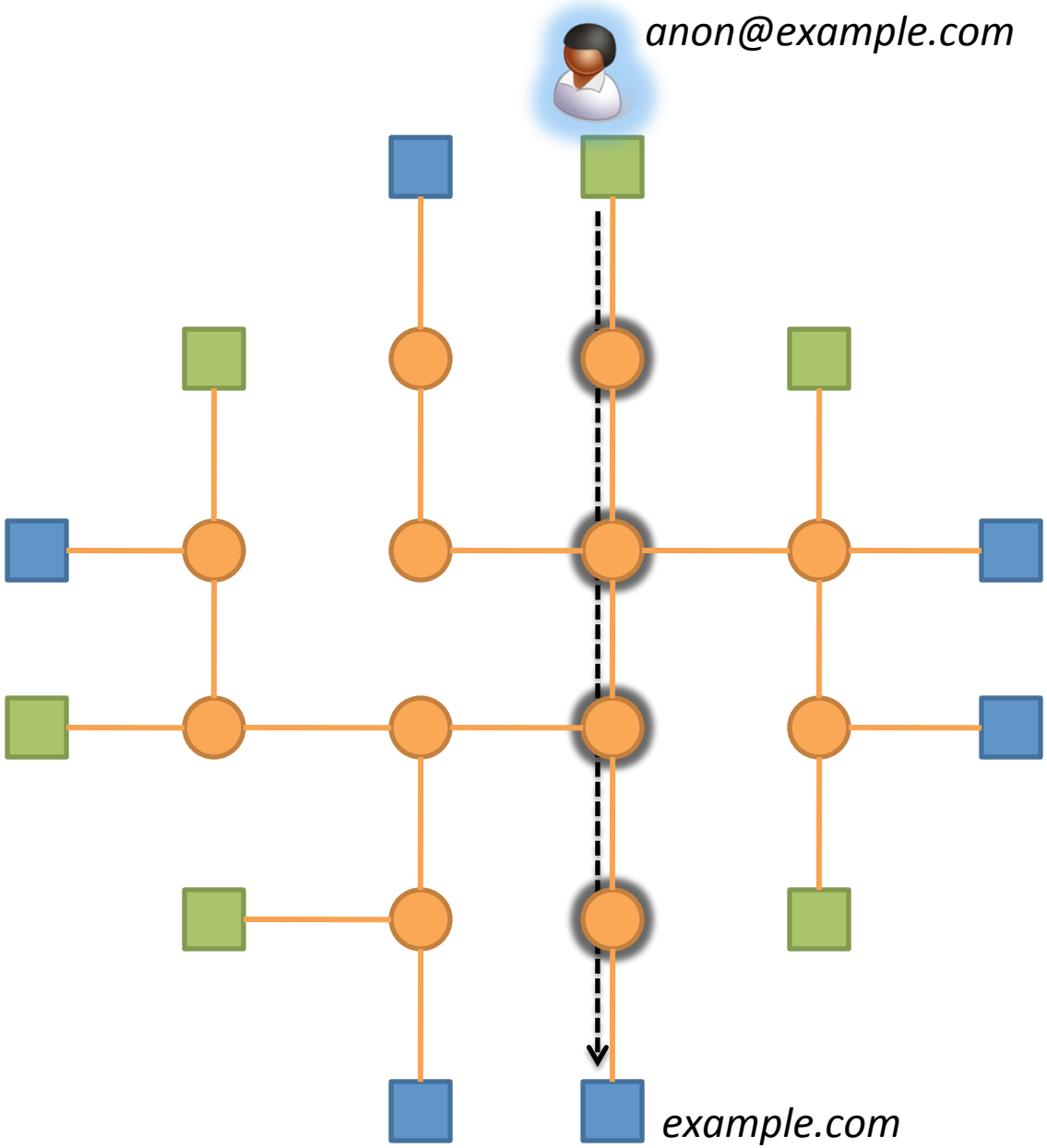
Example 3: AAA server cached state

User connected to service. Its AAA client obtains the server realm from the user's NAI.



Example 3: AAA server cached state

AAA client determines that it already has a non-expired key for the AAA server.



Example 3: AAA server cached state

AAA client establishes a RadSec connection with the AAA server, and the user's credentials are authenticated across this connection.