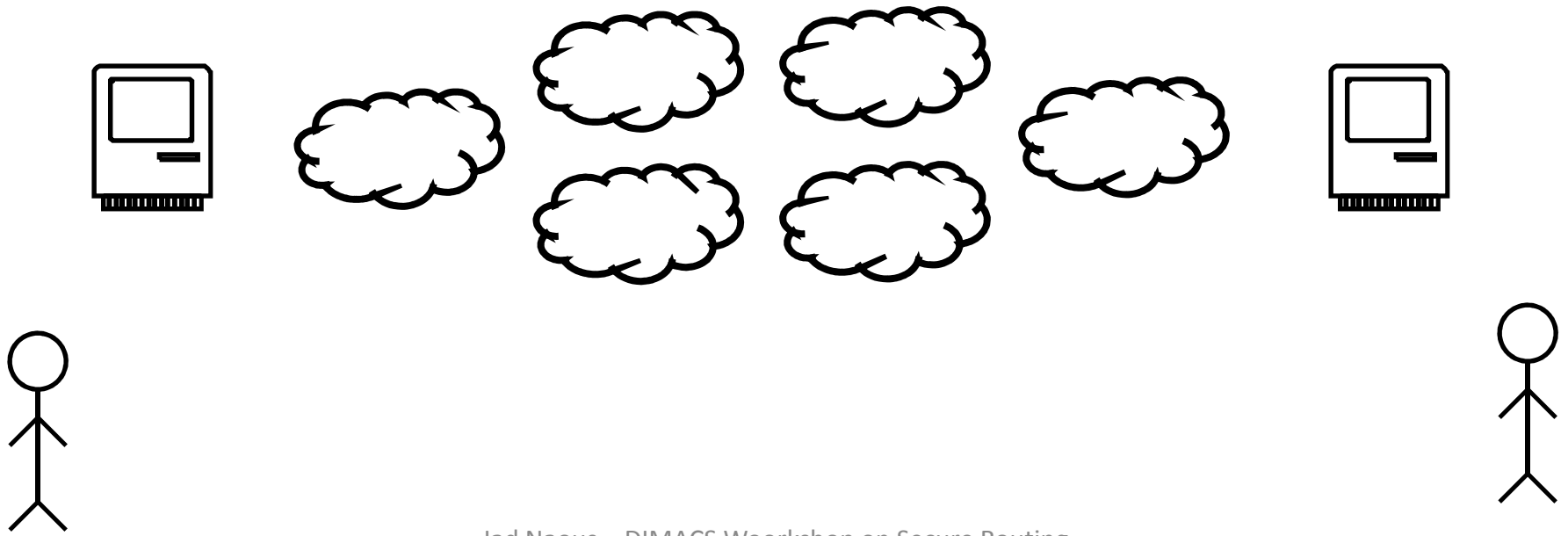
A Policy Framework for a Secure Future Internet

Jad Naous (Stanford University)
Arun Seehra (UT Austin)
Michael Walfish (UT Austin)
David Mazières (Stanford University)
Antonio Nicolosi (Stevens Institute of Tech)
Scott Shenker (UC Berkeley)

What do we want from the network?

Conflicting requirements

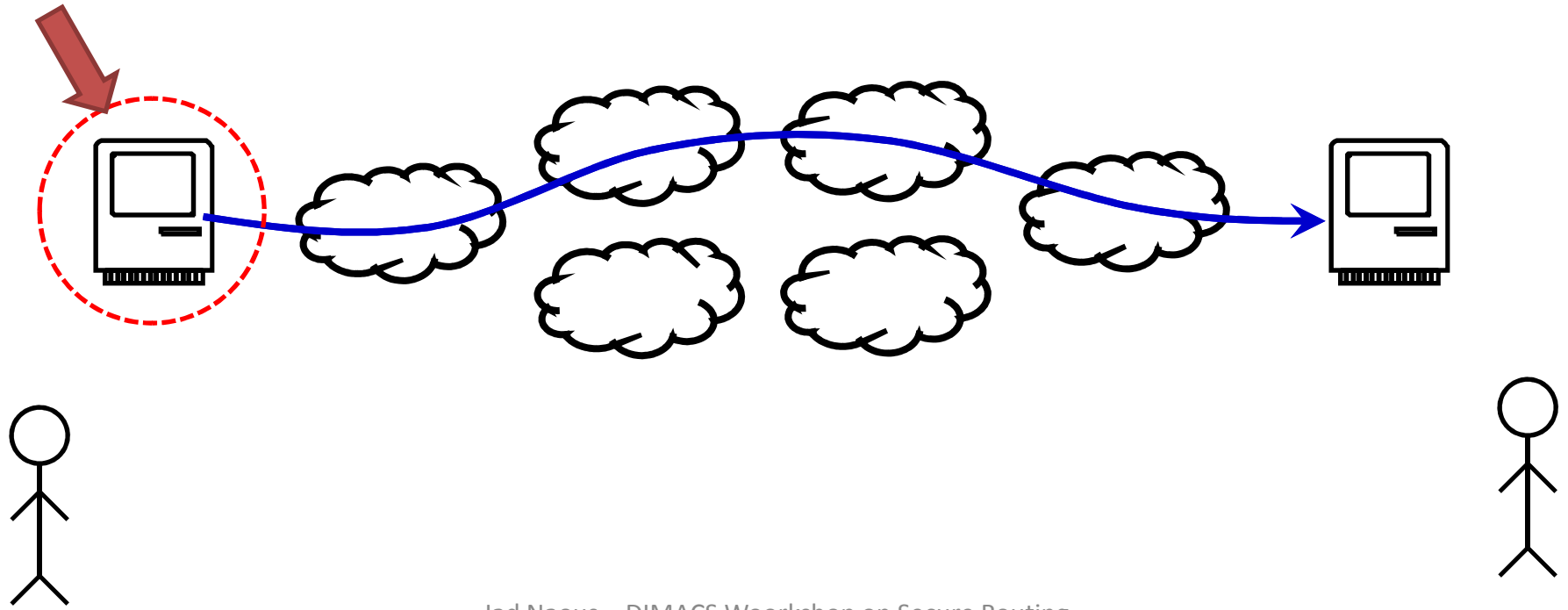
from many stakeholders



Network Policies

Conflicting requirements

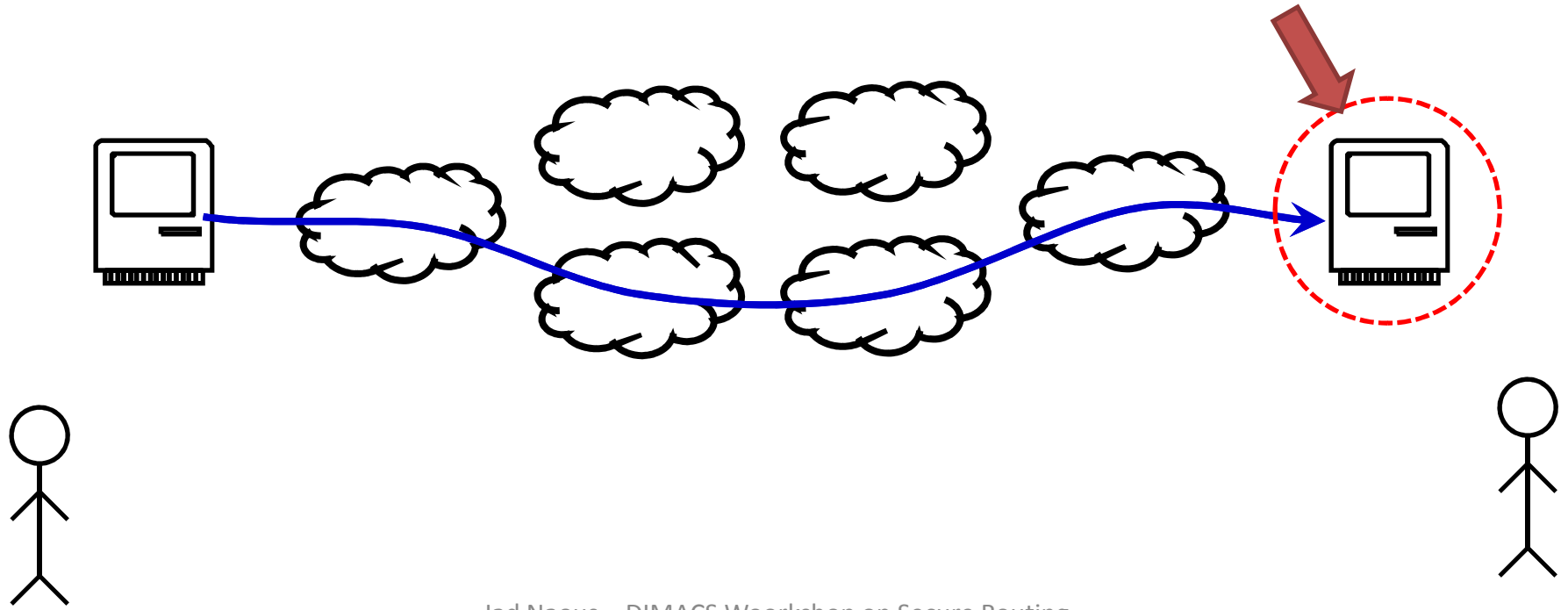
from many stakeholders



Network Policies

Conflicting requirements

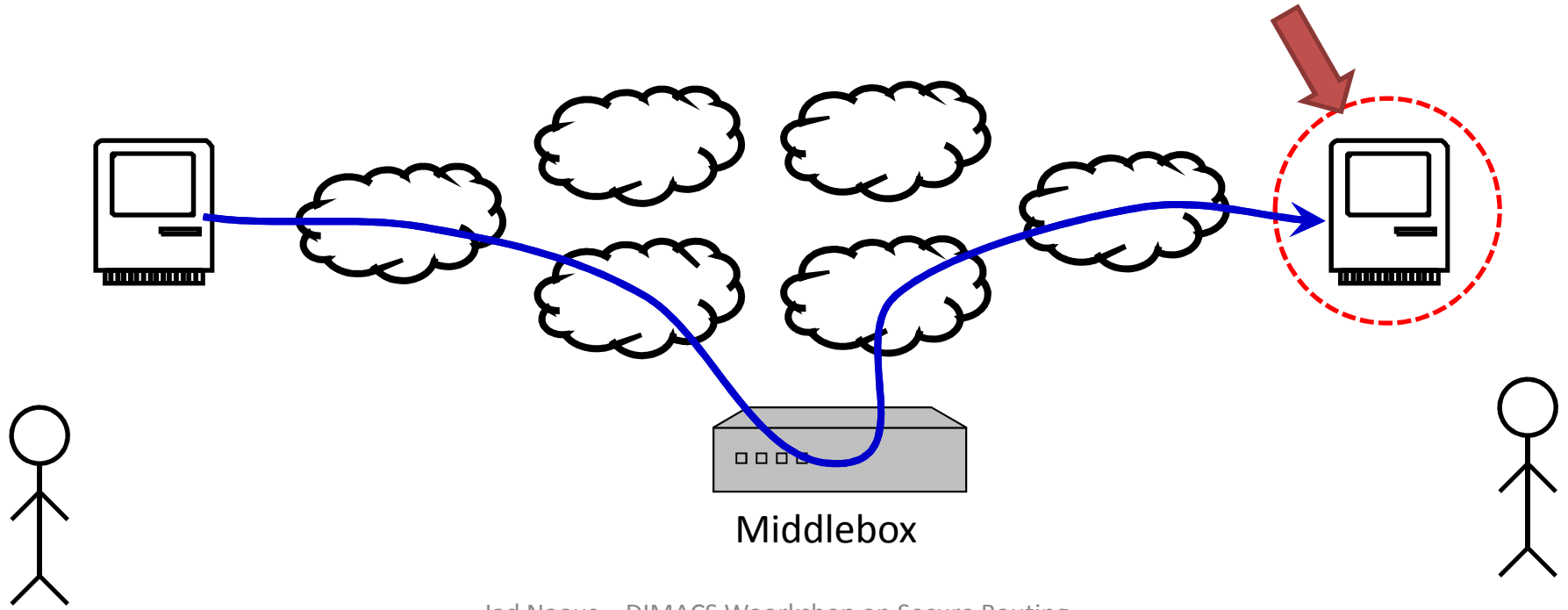
from many stakeholders



Network Policies

Conflicting requirements

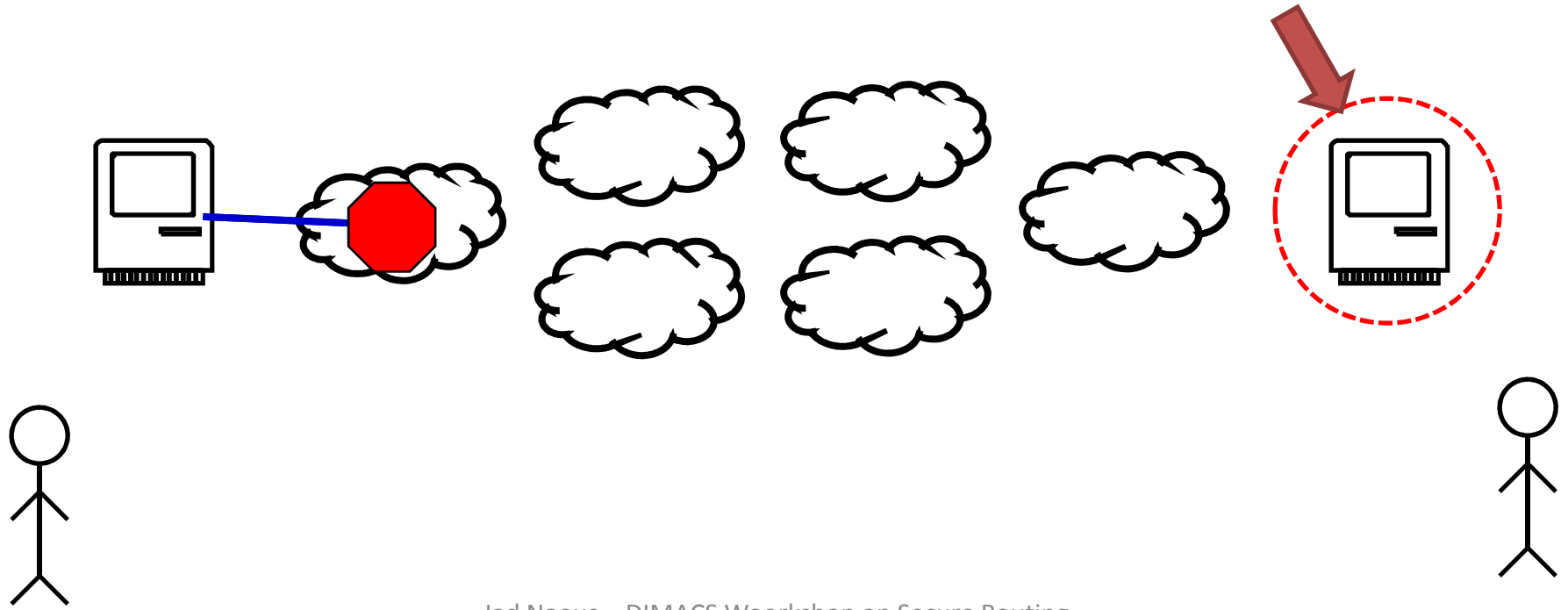
from many stakeholders



Network Policies

Conflicting requirements

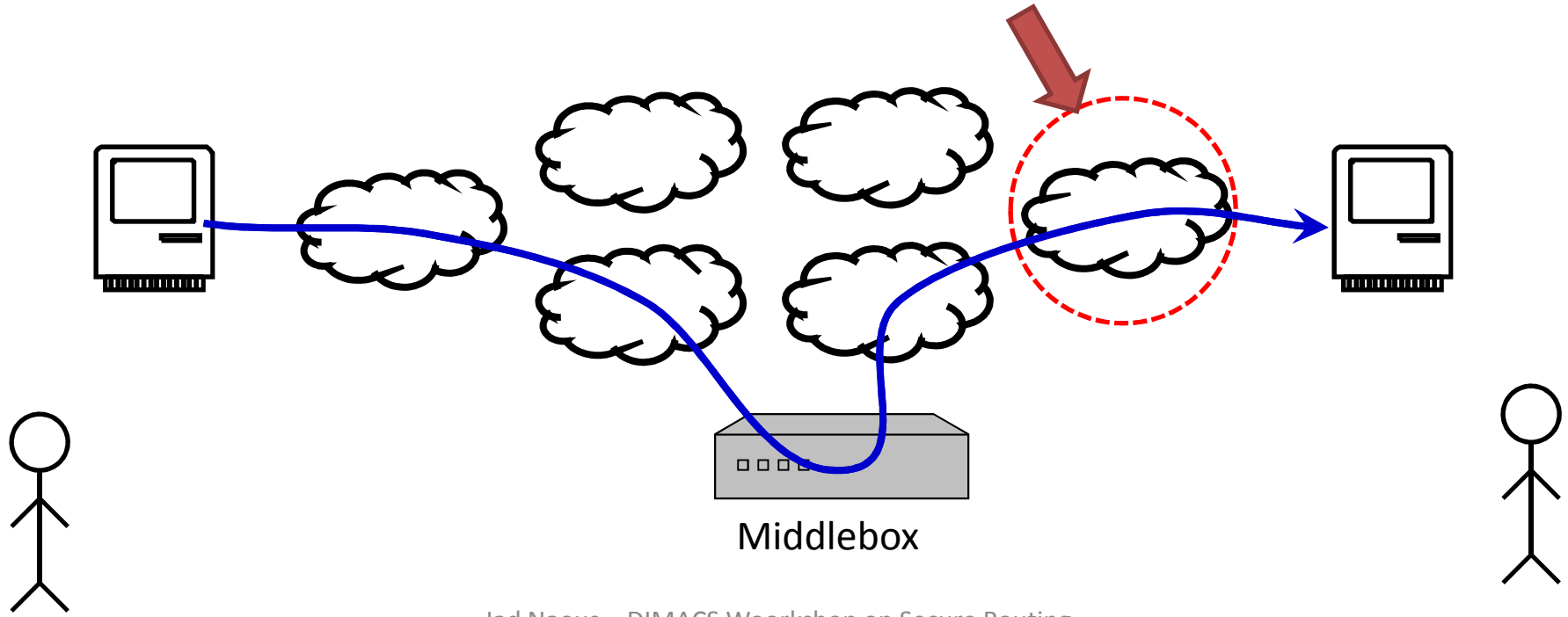
from many stakeholders



Network Policies

Conflicting requirements

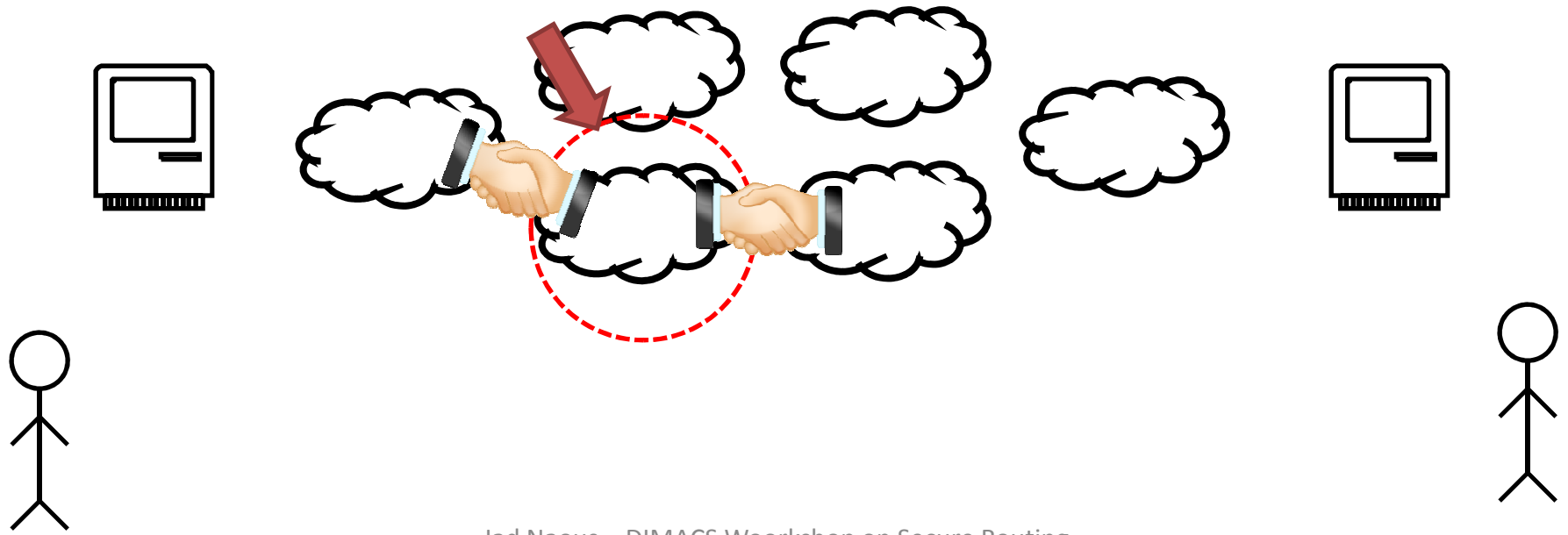
from many stakeholders



Network Policies

Conflicting requirements

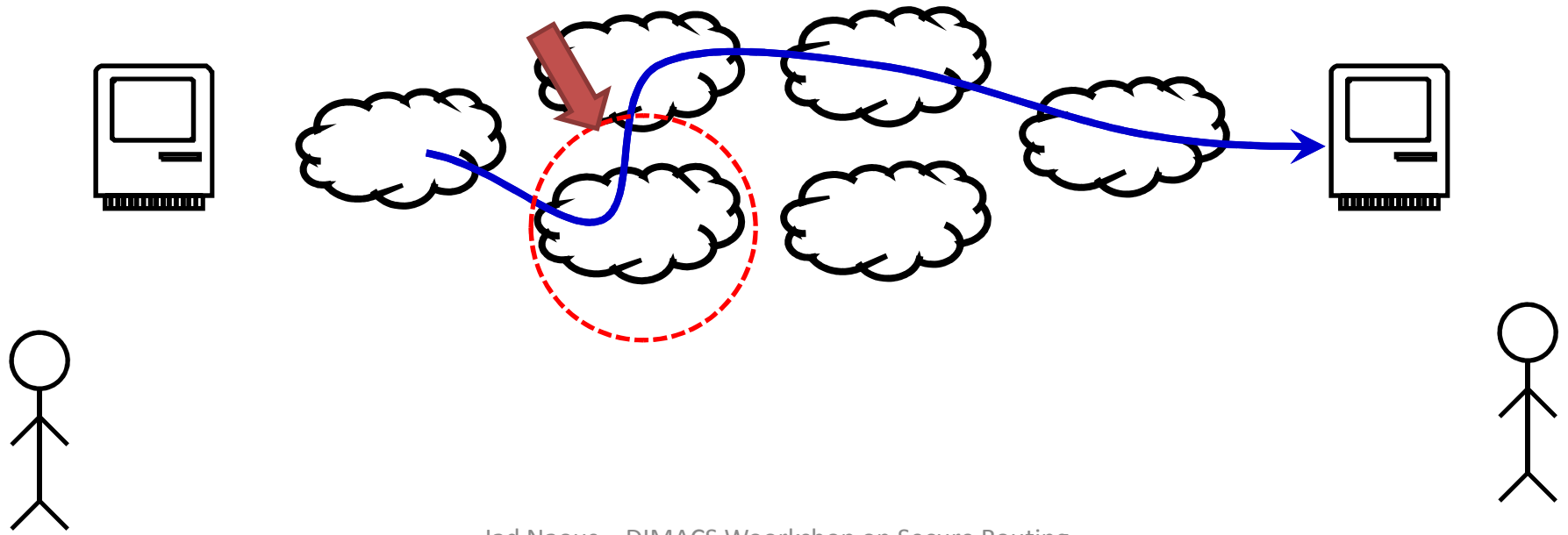
from many stakeholders



Network Policies

Conflicting requirements

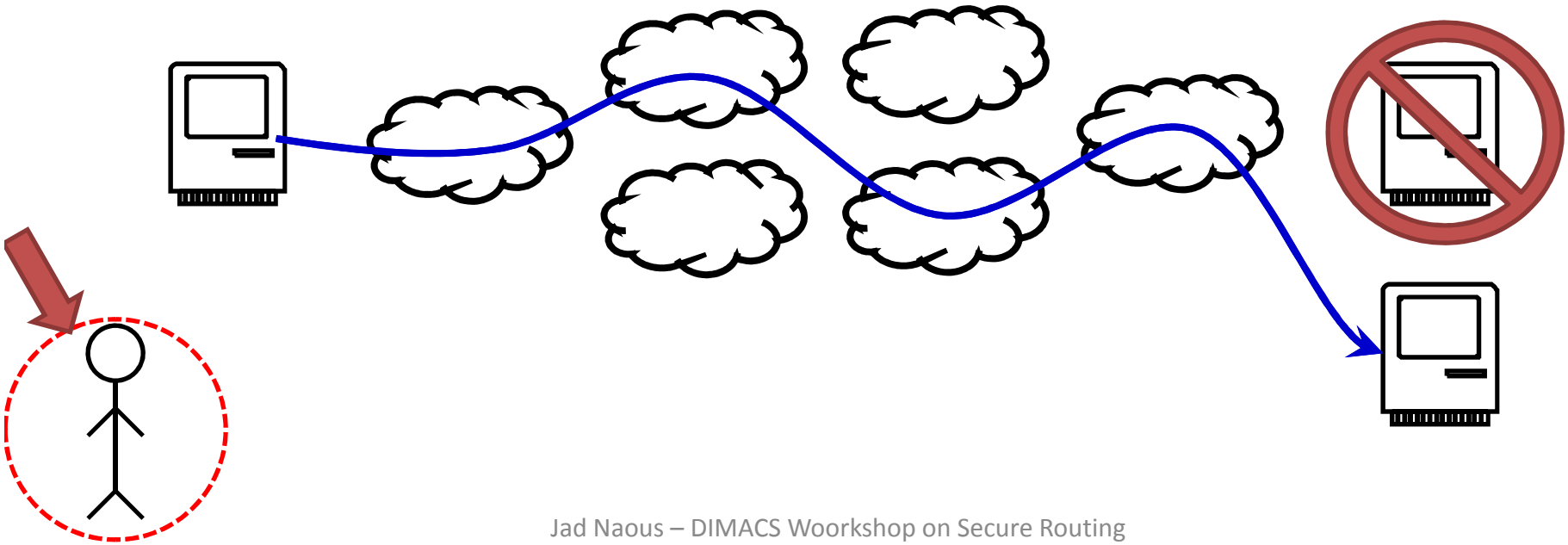
from many stakeholders



Network Policies

Conflicting requirements

from many stakeholders



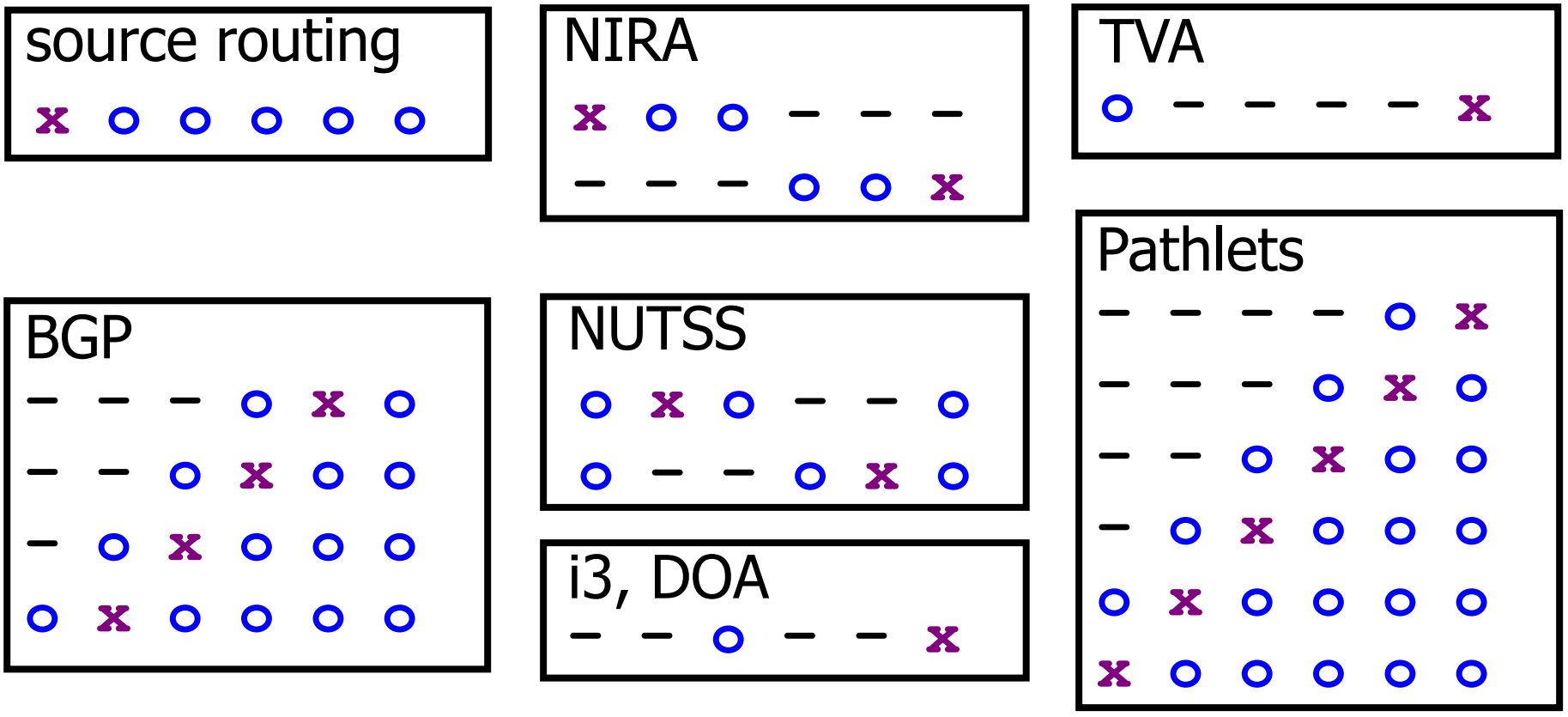
Network Policies

There are many stakeholders:

senders, receivers, enterprises that are both senders and receivers (e.g. data centers), service providers, security middlemen (à la Prolexic), governments, data owners, …

Each has many valid policy goals, and they might conflict.

Prior proposals:
Large union, small intersection



[legend: x exerts control over o's]

Prior Proposals

Incomplete or insufficient

Incompatible

What Types of Policies for the Future Internet?

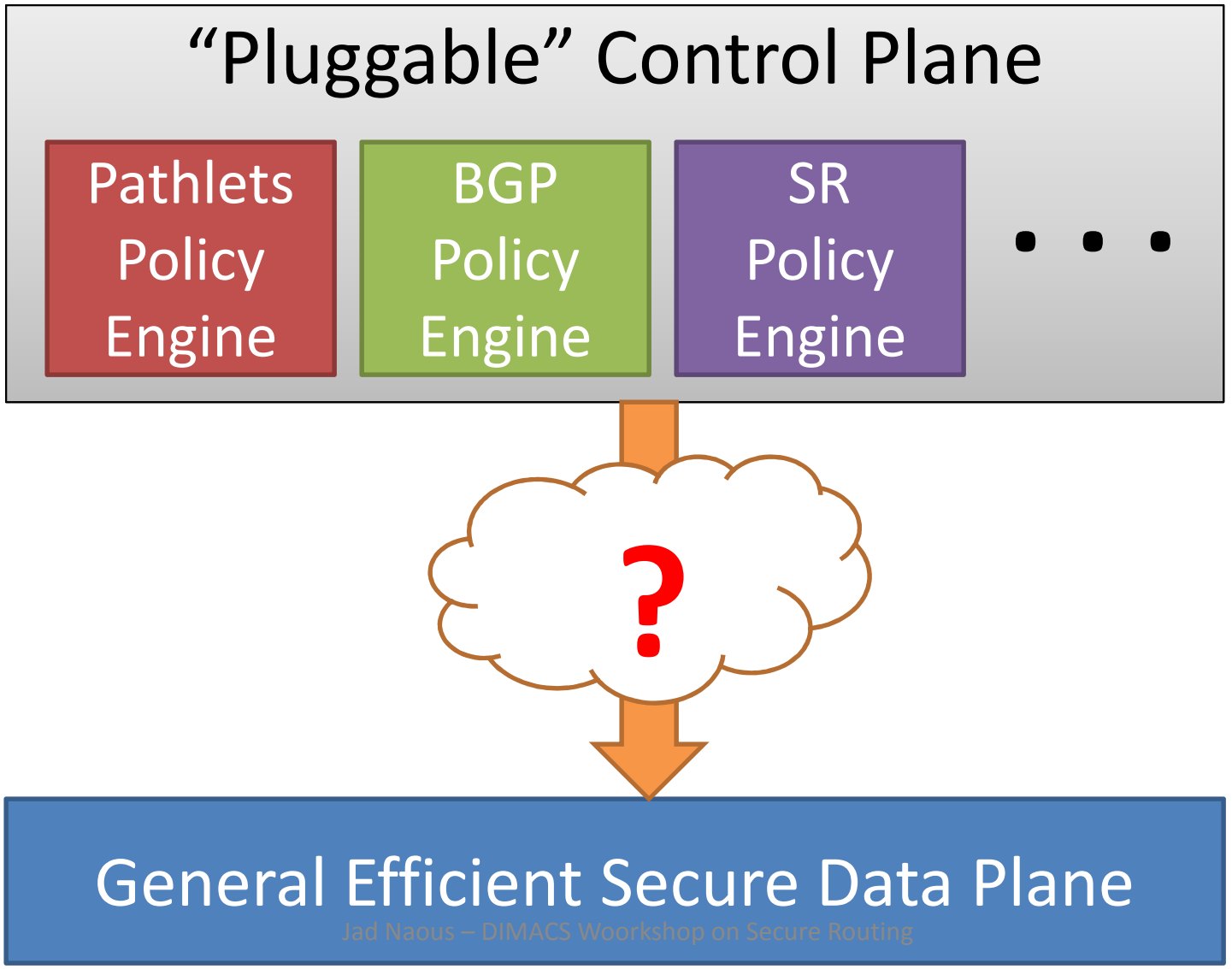
Three choices:

1. Embrace the status quo: *Do nothing.*
   Unsatisfactory.
2. Make a hard choice: *Select the "right" subset.*
   A high-stakes gamble.
3. Choose "all of the above": *Take union of controls.*
   Preserve all options; no picking winners/losers.

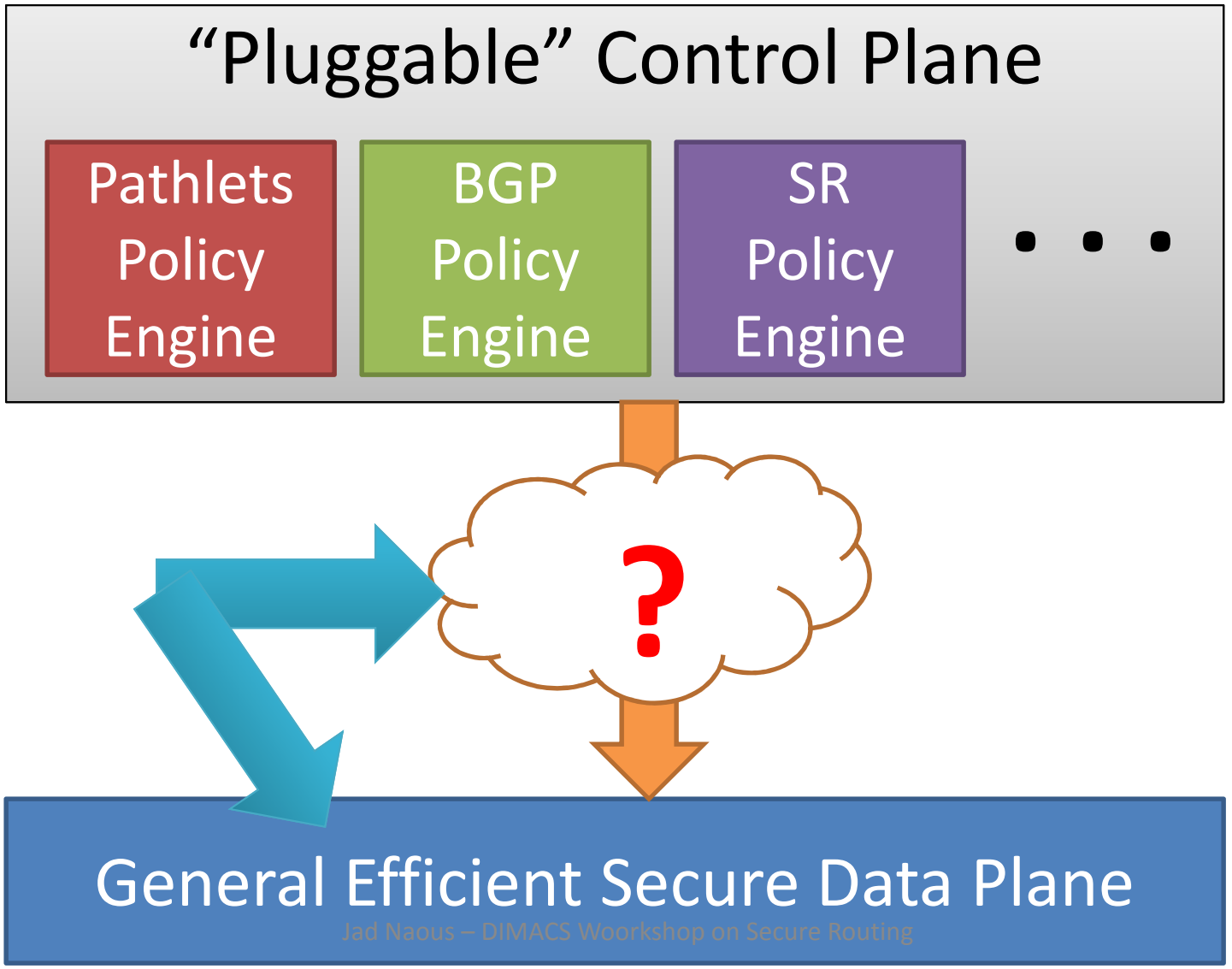
“All of the above” brings challenges:

1. How do we enable all these different policies?
2. How do we enforce all of them efficiently?

The ICING Policy Framework



The ICING Policy Framework



Outline

- How general is general?
  (What is the control? Who gets control? How can it be used?)
- How do we enforce policy decisions in the data plane?
- What is the control/data plane interface and how can it be used?

Outline

- How general is general?
  (What is the control? Who gets control? How can it be used?)
- How do we enforce policy decisions in the data plane?
- What is the control/data plane interface and how can it be used?

Control over what?

Policy requirements

$\Rightarrow$ Who handles the packets and how

$\Rightarrow$The path or parts of it (interdomain-level)

Control over what?

Policy requirements

$\Rightarrow$ Who handles packets they send/receive/transit and how

$\Rightarrow$ The path or parts of it (interdomain-level)

For most flexibility:

Give control over full path

Who gets control?

Three principles:

1. Entities whose network resources are consumed.
2. Entities that are consuming network resources.
3. Entities should be within a single layer – the network layer.

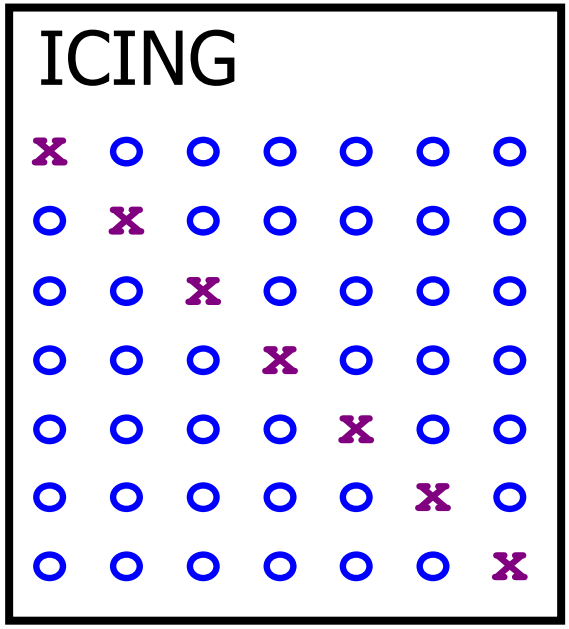
Who gets control?

The three principles

$\Rightarrow$ Give control to all entities on the path.

Other stakeholders use other layers or external power of authority (e.g. laws).

ICING's Policy Principle



A path is legal if and only if
all participants on the path approve of the path.

Architecture enforces that only legal paths are used.

How general are policies?

- Provider: Allow use of high speed links from 5pm to 8am only
- Internet2: Only carry traffic between universities
- Sender: Only use paths that my neighbor is using.

=> Policies can be arbitrary.

For flexibility and evolvability:

Allow arbitrary policies

For accuracy:

Provide sufficient information

What are policy decisions based on?

1. The path

2. Consumed resources:
   - Long/short haul, high/low speed, transit/delivery, …

3. Arbitrary external information:
   - Billing status, costs, time of day
   - Does everyone else consent?

Checkpoint Summary

- There are many stakeholders in a communication, and we give control to all network-level participants.
- For most flexibility and to satisfy the largest number of requirements we need to give them control over the full path.
- For evolvability and flexibility, allow arbitrary policies and provide sufficient information

Outline

- How general is general?
  (What is the control? Who gets control? How can it be used?)
- How do we enforce policy decisions in the data plane?
- What is the control/data plane interface and how can it be used?

Secure Routing Insufficient

Data packets today do not necessarily follow BGP-given routes

i.e. Data plane does not necessarily conform to the control plane.

Challenges

Many challenges:

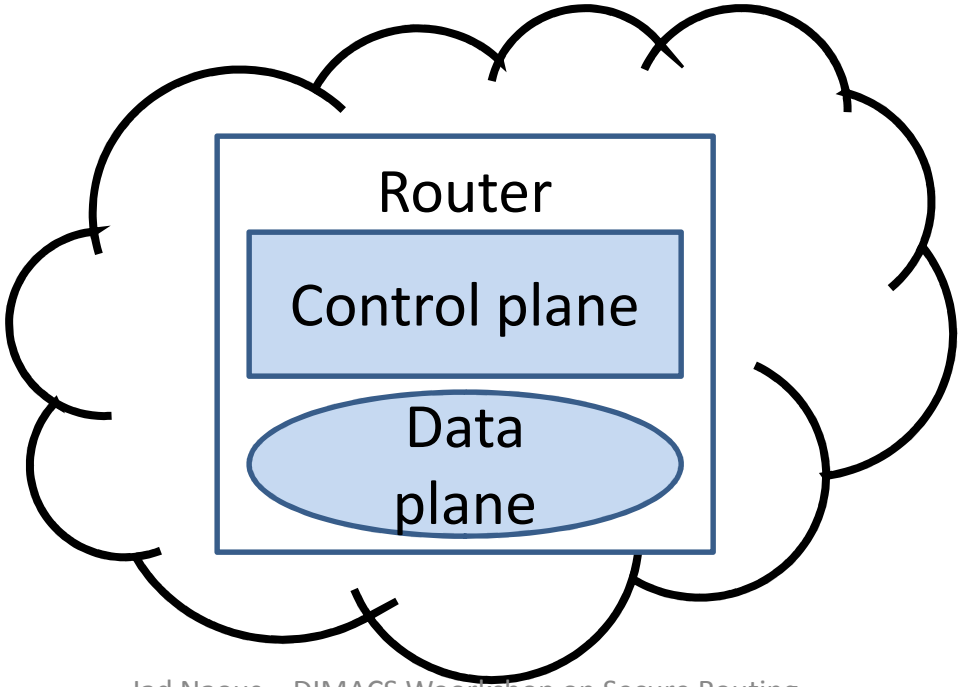
- Enabling arbitrary informed policies
- Enforcing policy decisions at line-rate
- Handling errors and network failures in a locked-down Internet
- Delegating access
- Bootstrapping

Challenges

Many challenges:

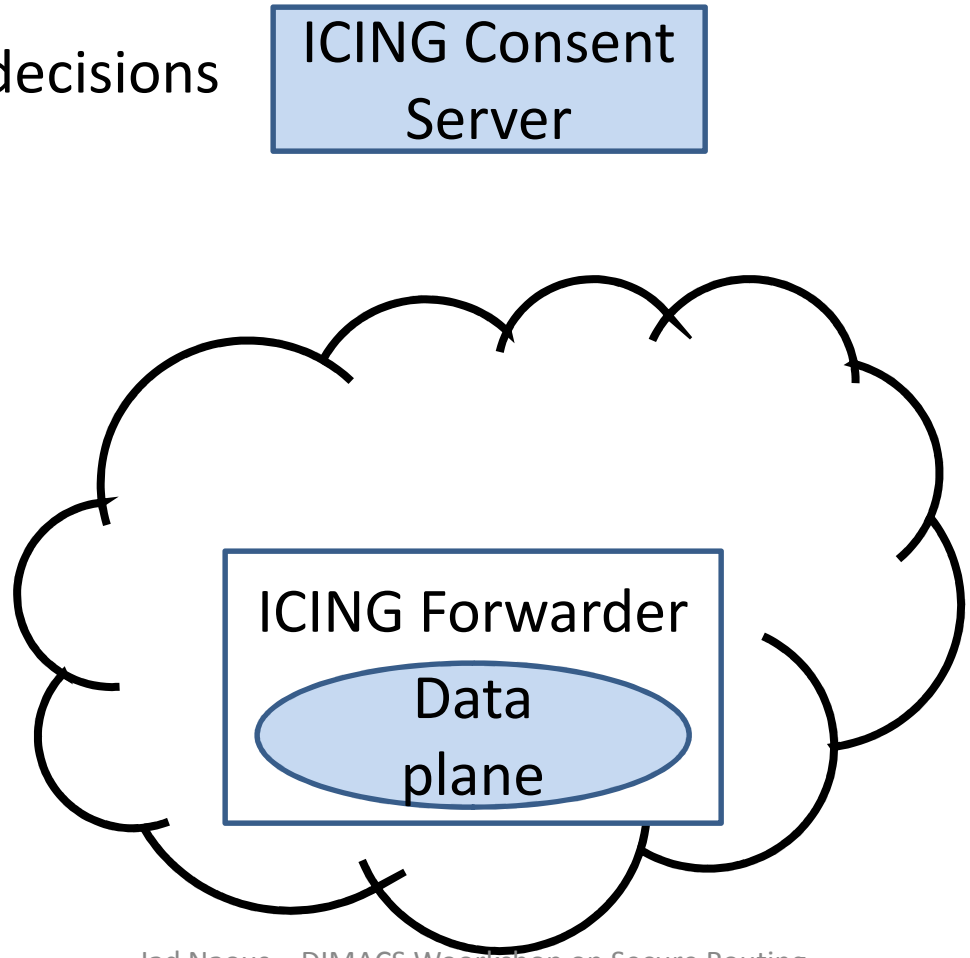
- Enabling arbitrary informed policies
- Enforcing policy decisions at line-rate
- Handling errors and network failures in a locked-down Internet
- Delegating access
- Bootstrapping

Challenge:
Enabling arbitrary informed policies



Challenge:
Enabling arbitrary informed policies

Makes all policy decisions



Enforces policy decisions

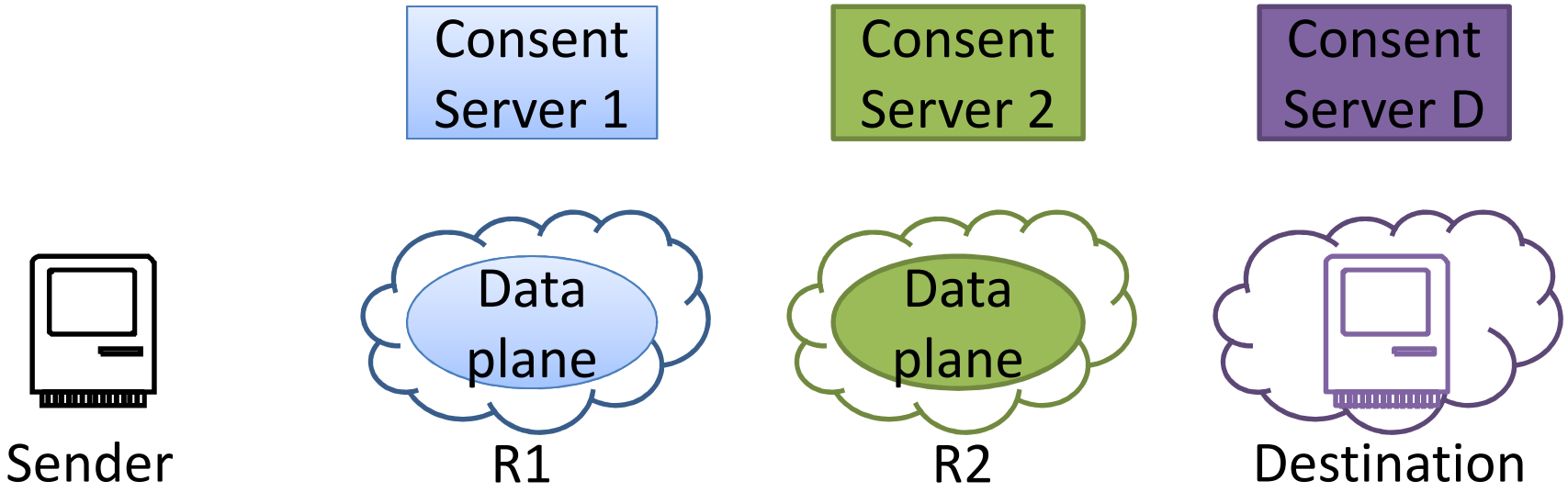
Challenge:
Enforcing policy decisions at line-rate

1. Make sure that the path is legal

2. Make sure that the path is followed

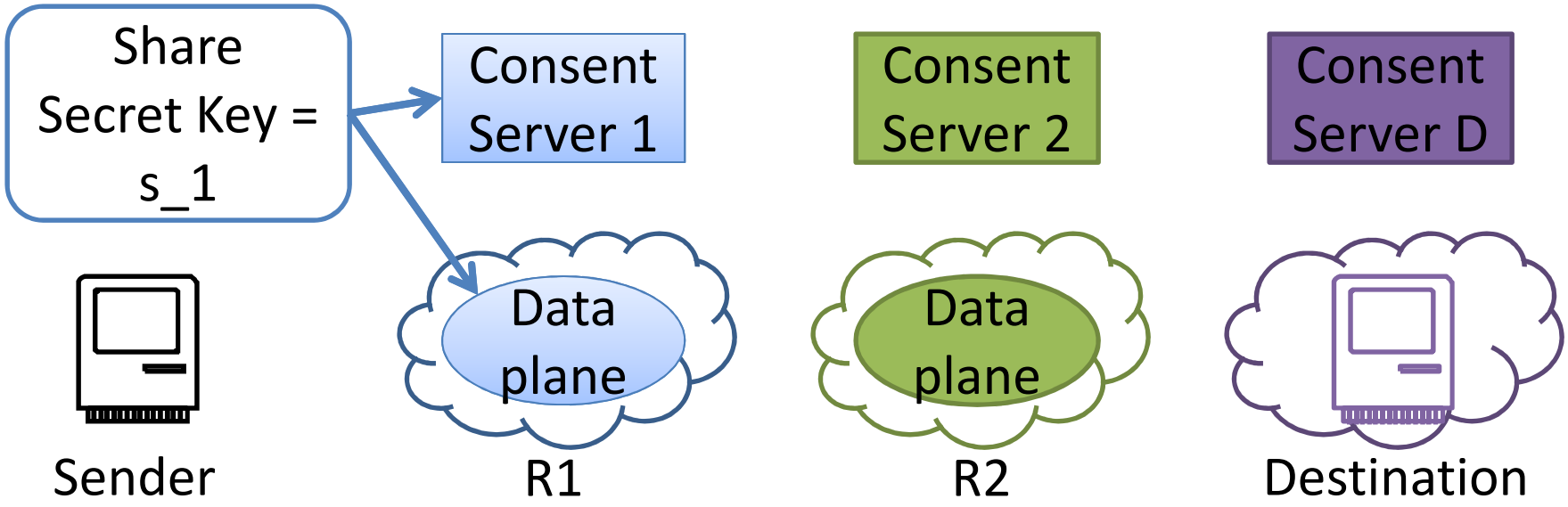
Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



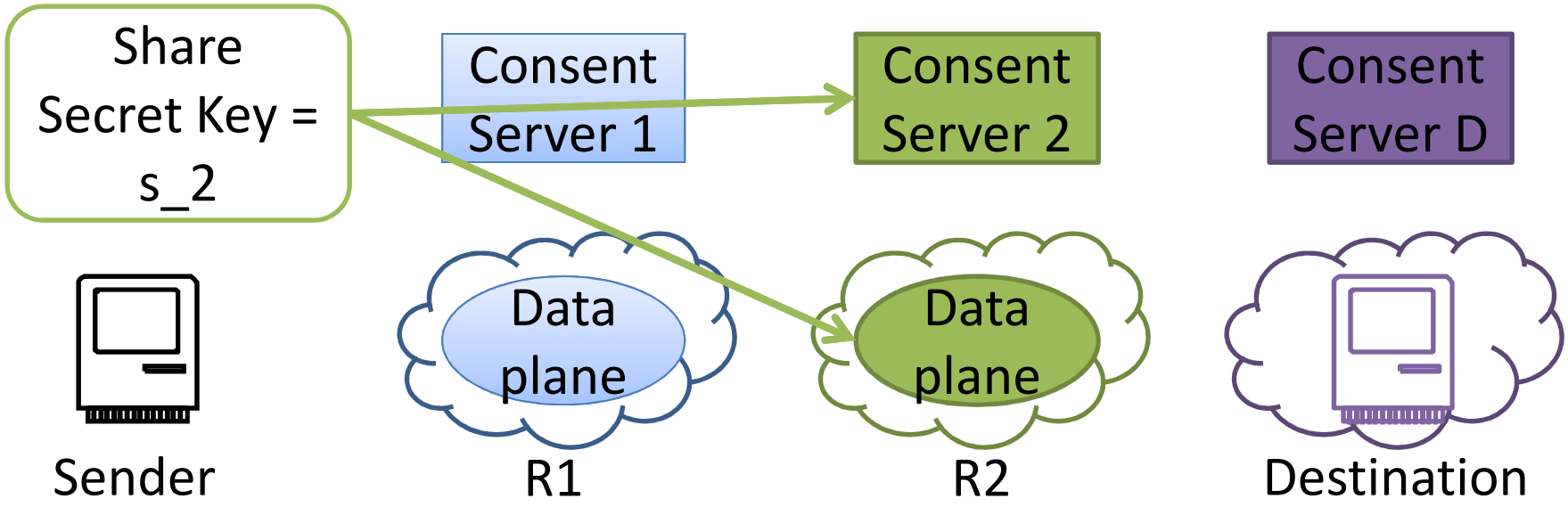
Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



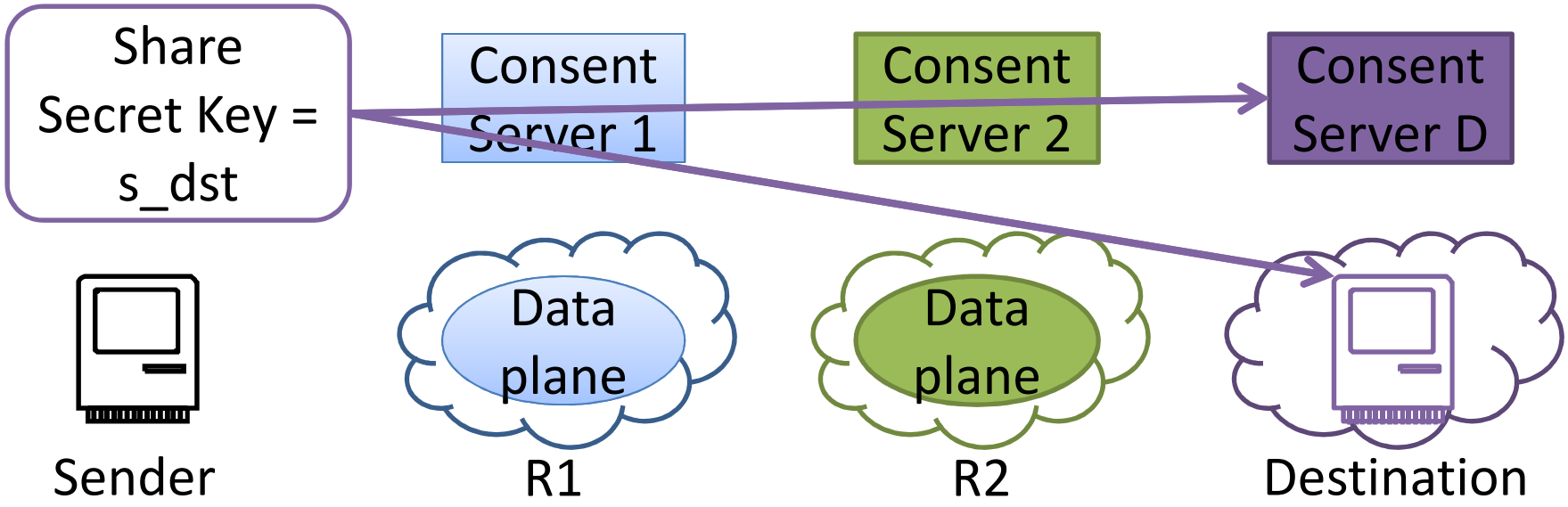
Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



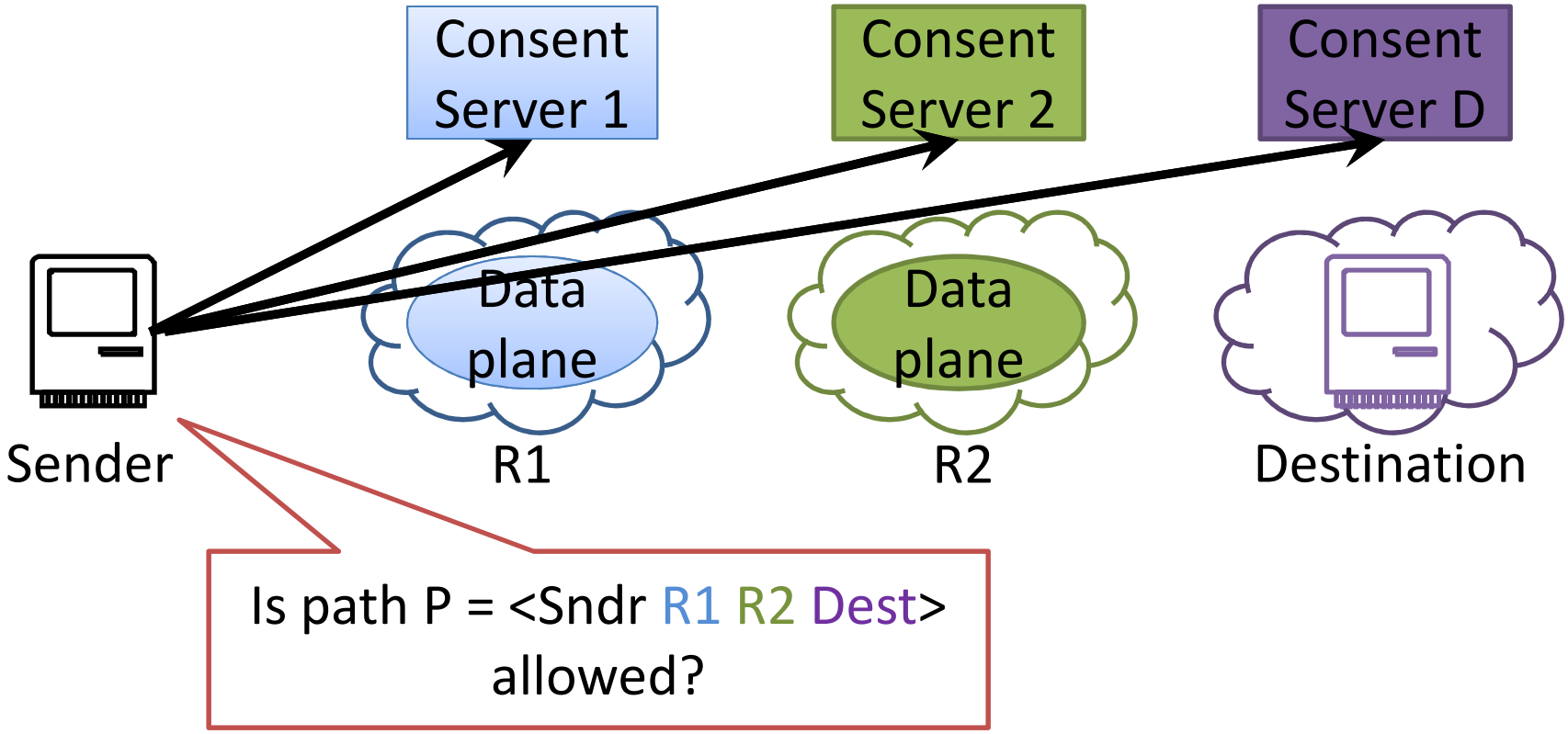
Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



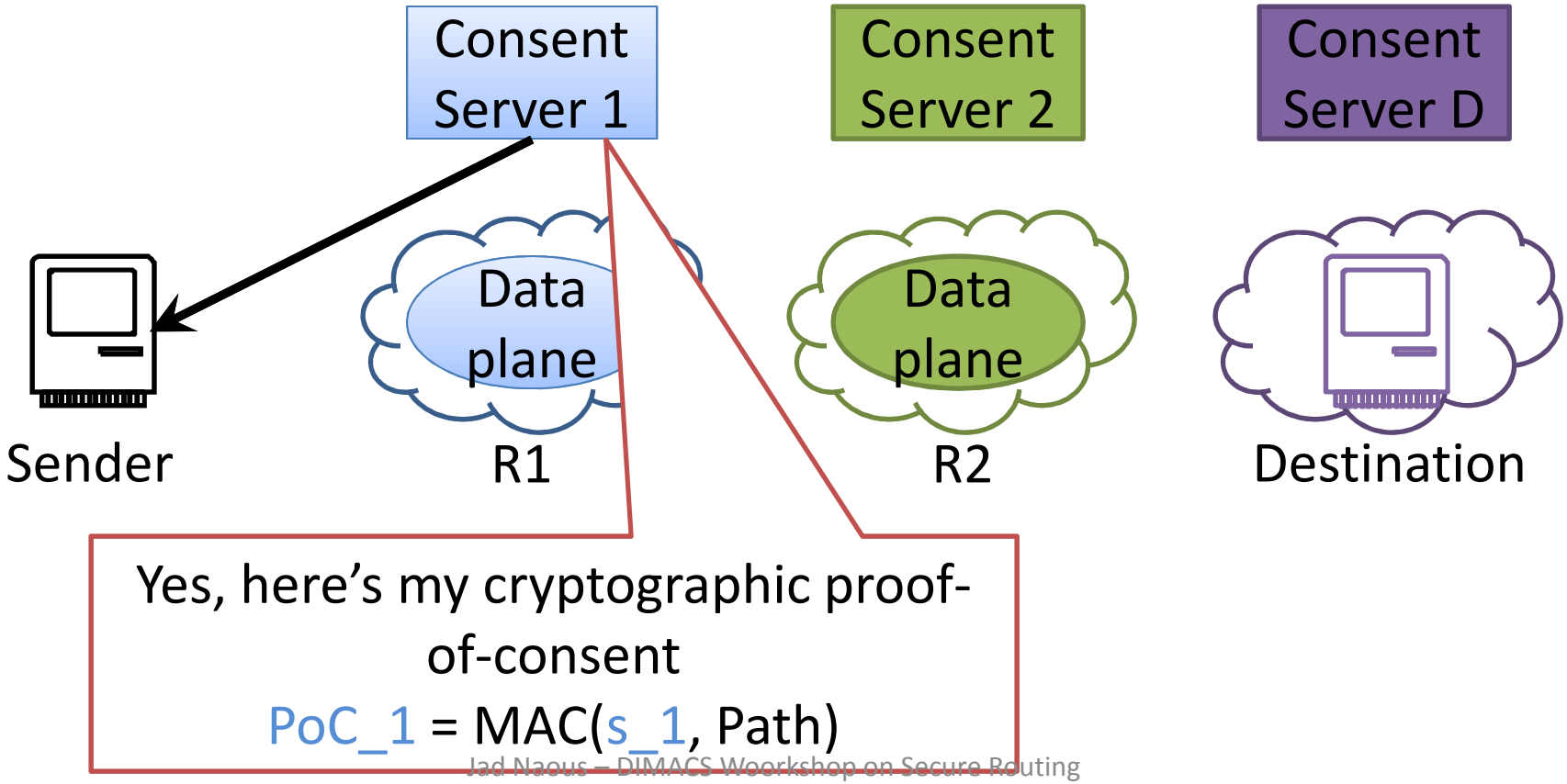
Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



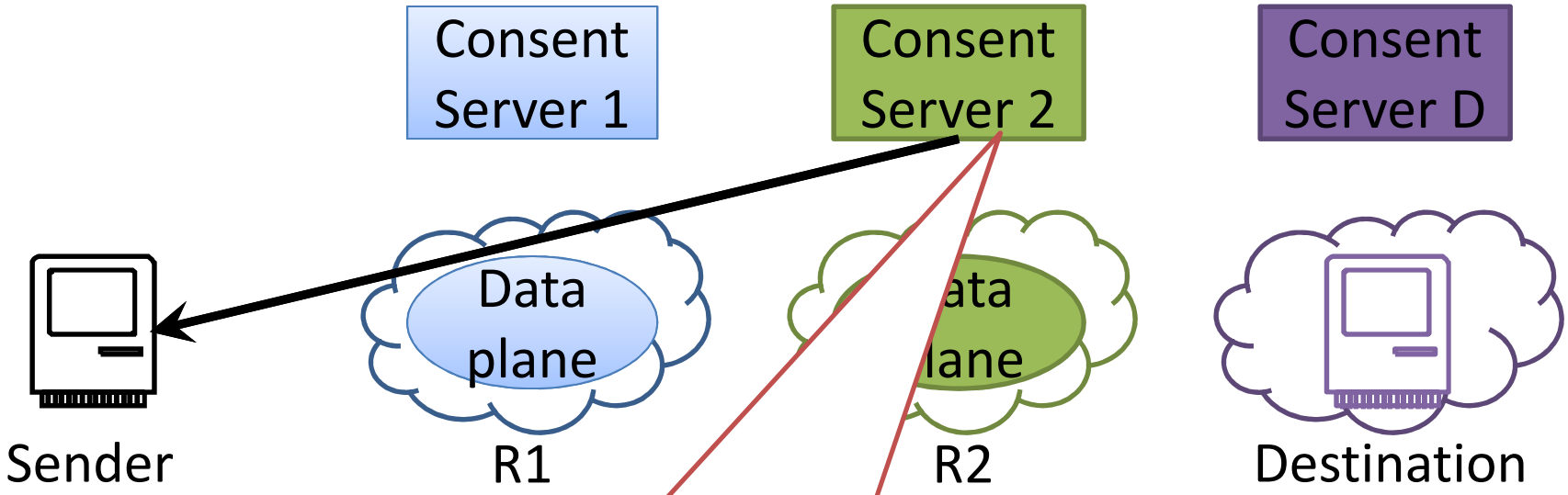
Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



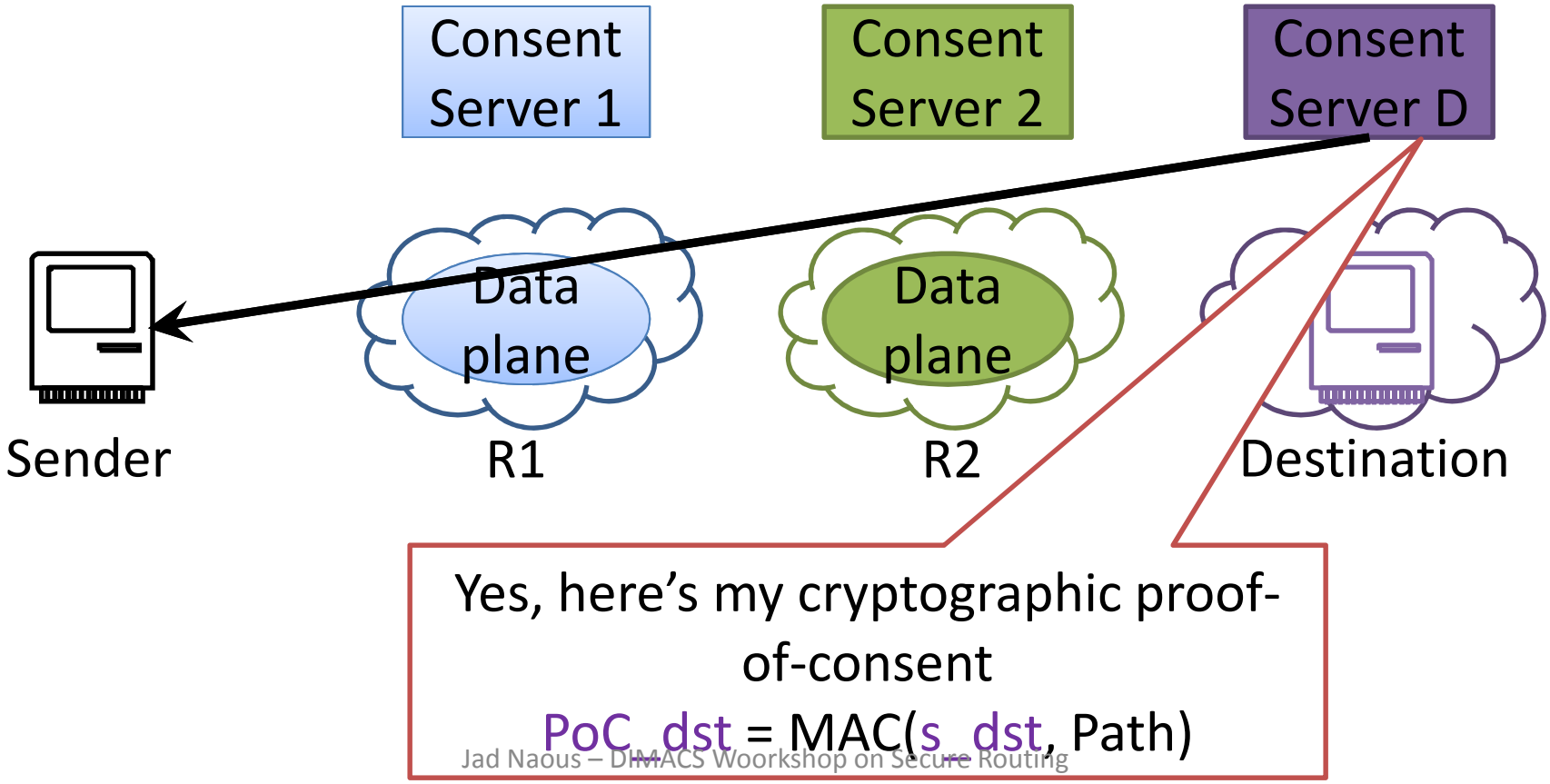
Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



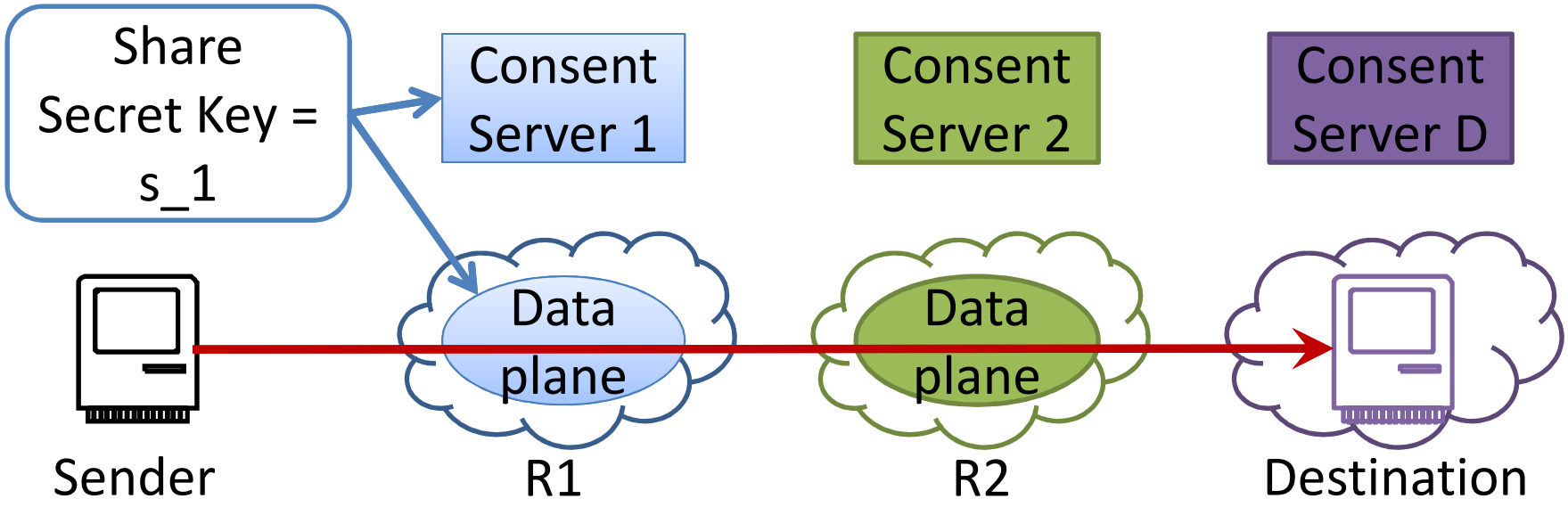
Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



Challenge:
Enforcing policy decisions at line-rate

Step 1: Make sure the path is legal



Packet =
<Path, PoC_1, PoC_2, PoC_dst, data>
PoCs verifiable by data plane using
Shared secret keys s_1, s_2, s_dst

Challenge:
Enforcing policy decisions at line-rate

Notes:

1. Policy decisions made off the critical path
   - Once per path
     (not per-packet, not even per flow)
   - Before packet flow
2. Decision is encoded in cryptographic proof of consent using shared symmetric key.
3. Forwarders can verify that the consent server had approved of the path.

Challenge:
Enforcing policy decisions at line-rate

1. Make sure that the path is legal

2. Make sure that the path is followed

Challenge:
Enforcing policy decisions at line-rate

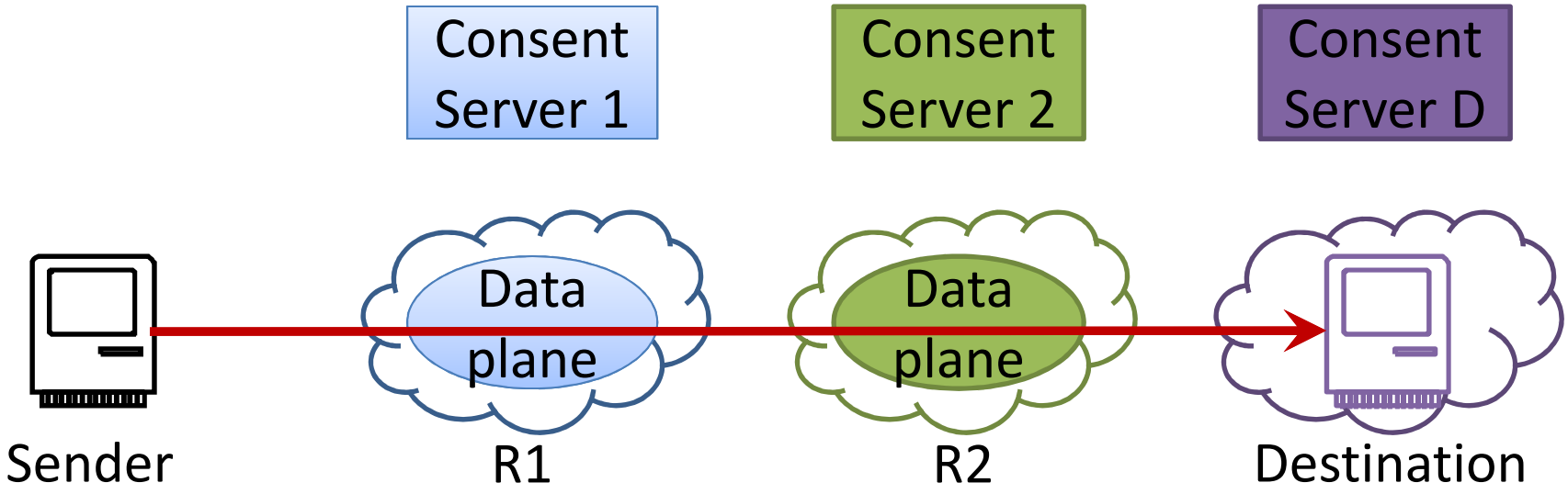
Step 2: Make sure the path is followed

- Problems:
  - Backbone speeds preclude digital signatures or public key crypto on the fast path.
  - Federated nature of the Internet precludes central root of trust, pre-configured shared secrets, etc…

- ICING overcomes these hurdles with new packet authentication techniques.

Challenge:
Enforcing policy decisions at line-rate

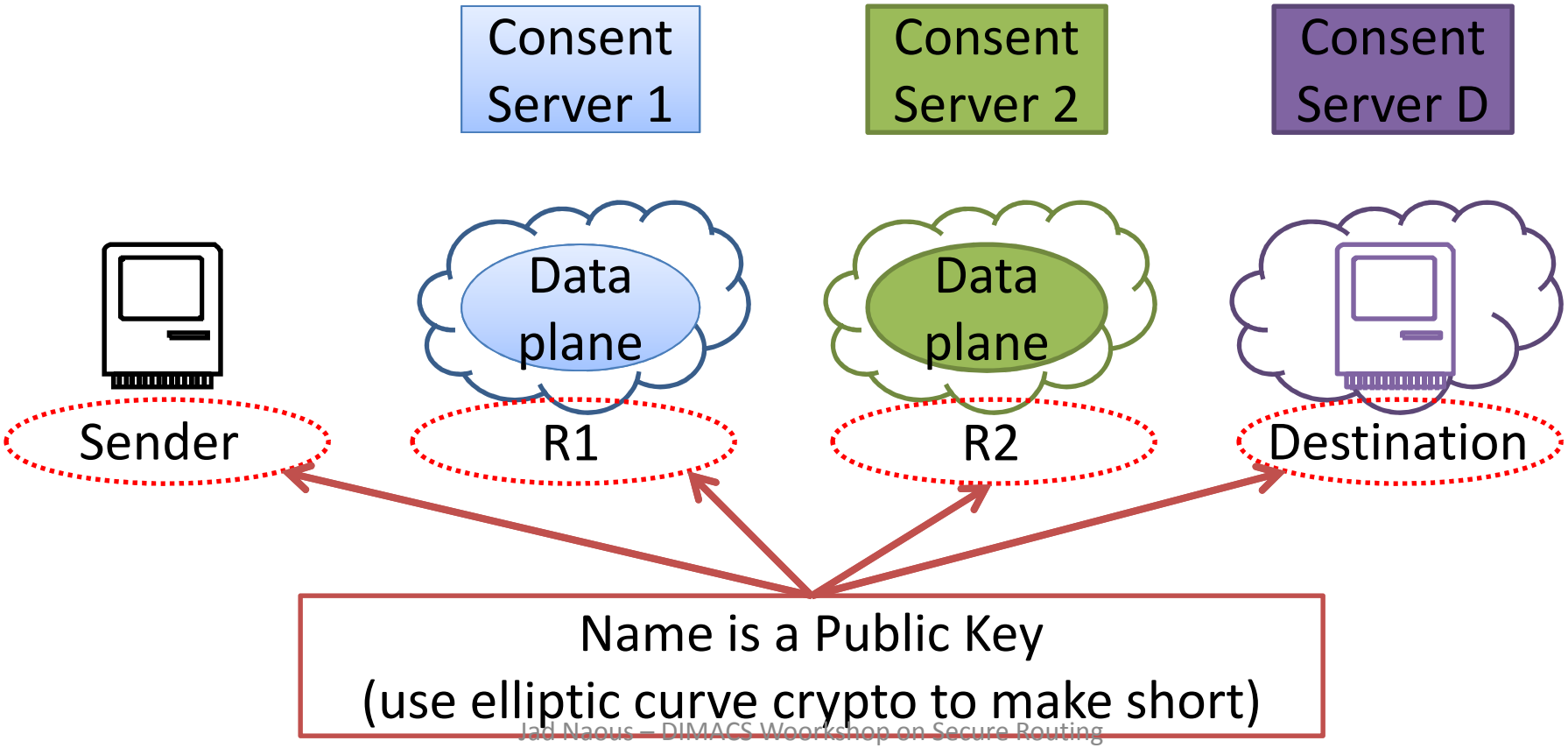
Step 2: Make sure the path is followed



Packet =
<Path, PoC_1, PoC_2, PoC_dst, V_1, V_2, V_dst, data>
V_i proves to Realm i that everyone before it has seen the packet.

Challenge:
Enforcing policy decisions at line-rate

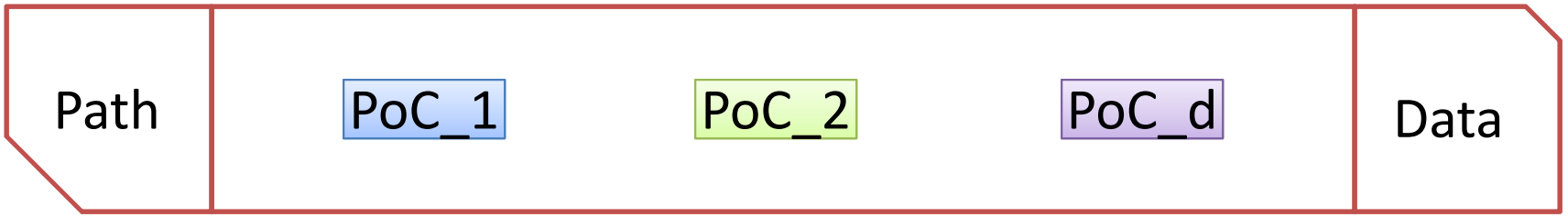
Step 2: Make sure the path is followed



Challenge:
Enforcing policy decisions at line-rate

Step 2: Make sure the path is followed

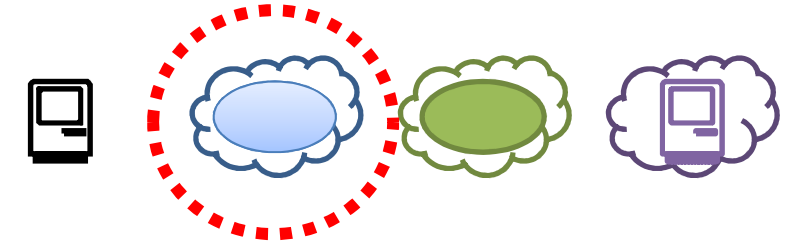
1. Verify consent & provenance
2. Prove provenance

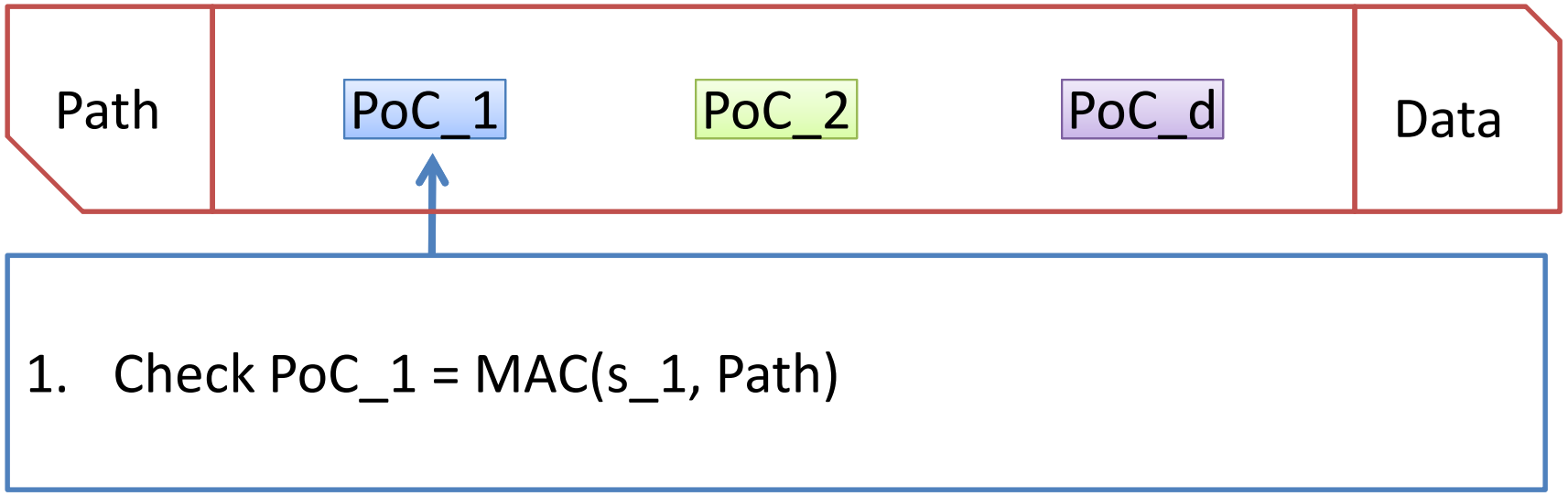


Challenge:
Enforcing policy decisions at line-rate

Step 2: Make sure the path is followed

1. Verify consent & provenance
2. Prove provenance

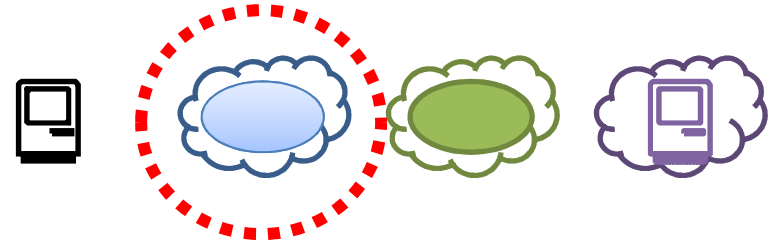


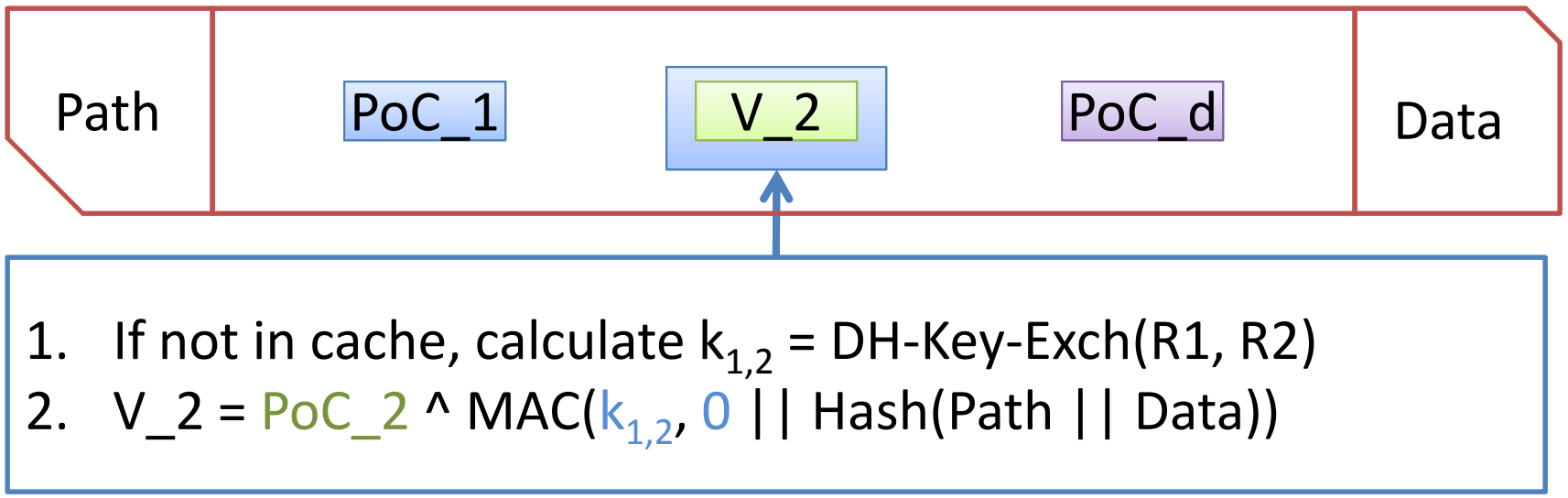


Challenge:
Enforcing policy decisions at line-rate

Step 2: Make sure the path is followed

1. Verify consent & provenance
2. Prove provenance

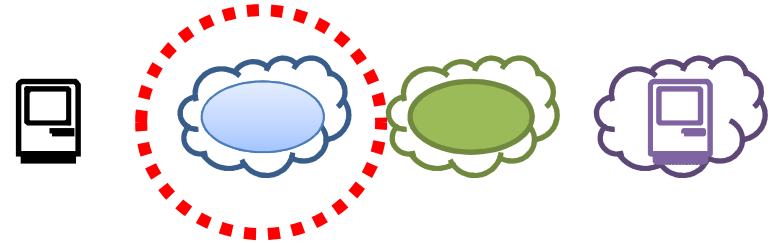


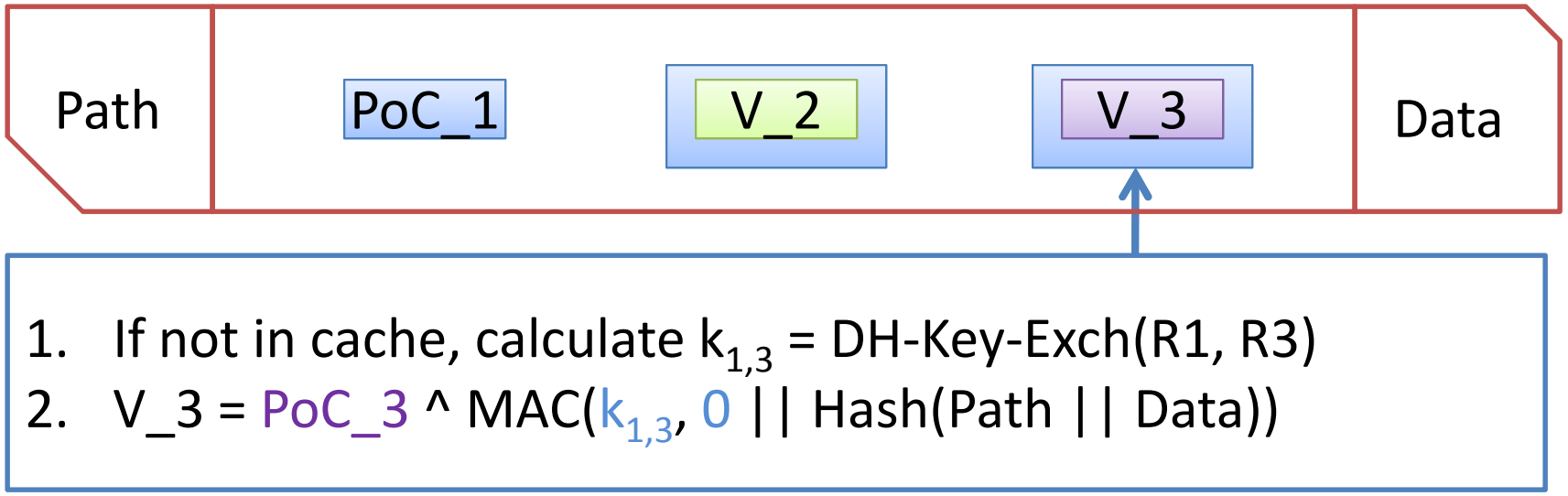


Challenge:
Enforcing policy decisions at line-rate

Step 2: Make sure the path is followed

1. Verify consent & provenance
2. Prove provenance

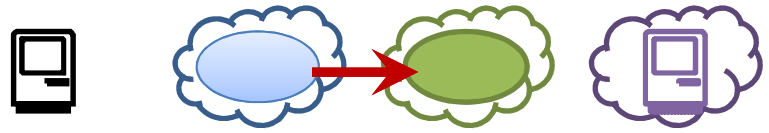


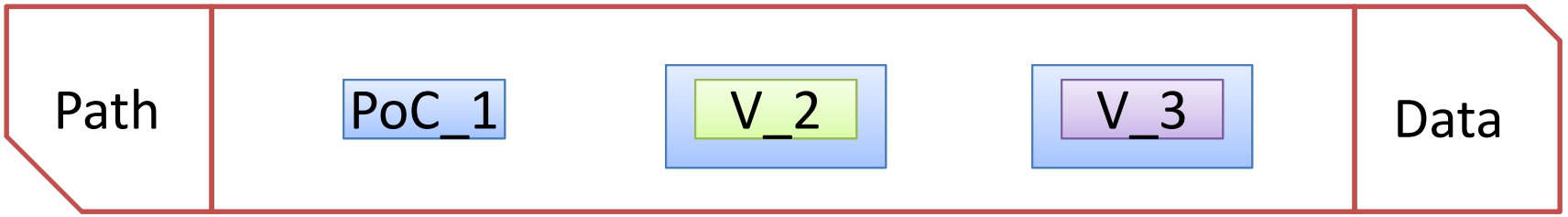


Challenge:
Enforcing policy decisions at line-rate

Step 2: Make sure the path is followed

1. Verify consent & provenance
2. Prove provenance

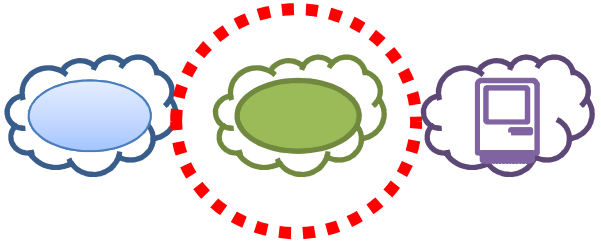


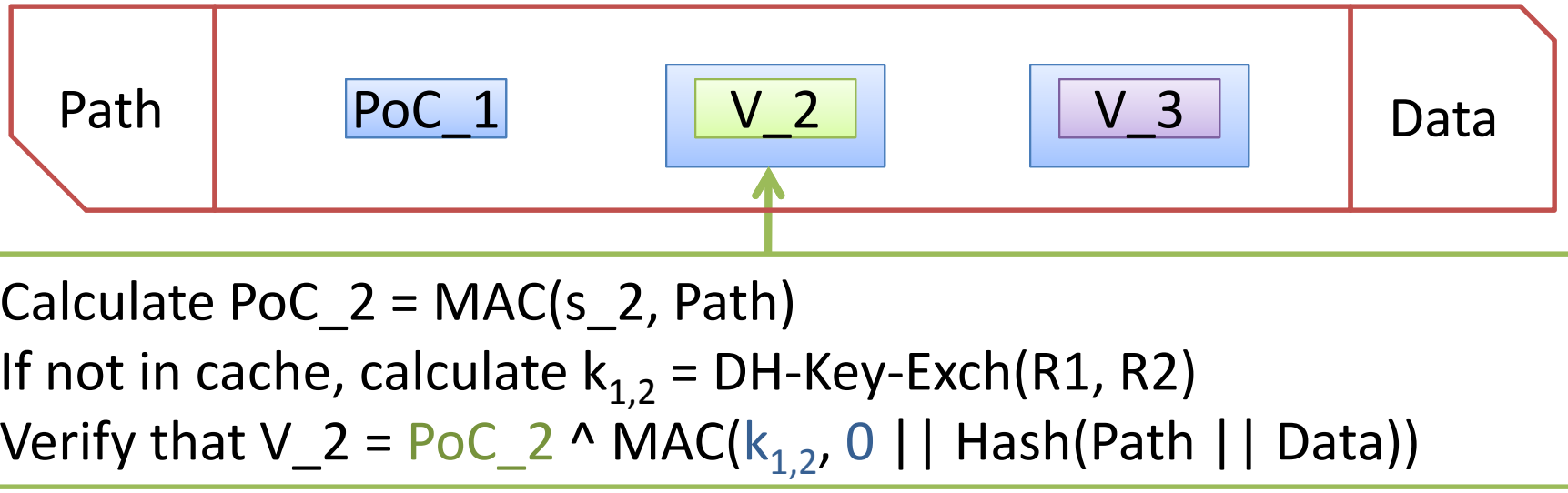


Challenge:
Enforcing policy decisions at line-rate

Step 2: Make sure the path is followed

1. Verify consent & provenance
2. Prove provenance

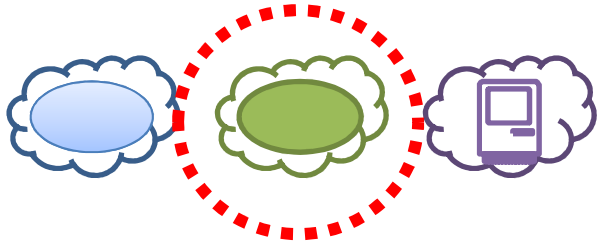


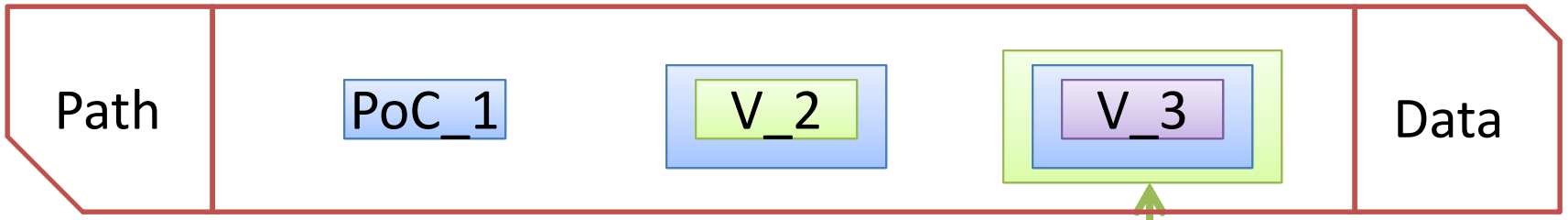


Challenge:
Enforcing policy decisions at line-rate

Step 2: Make sure the path is followed

1. Verify consent & provenance
2. Prove provenance



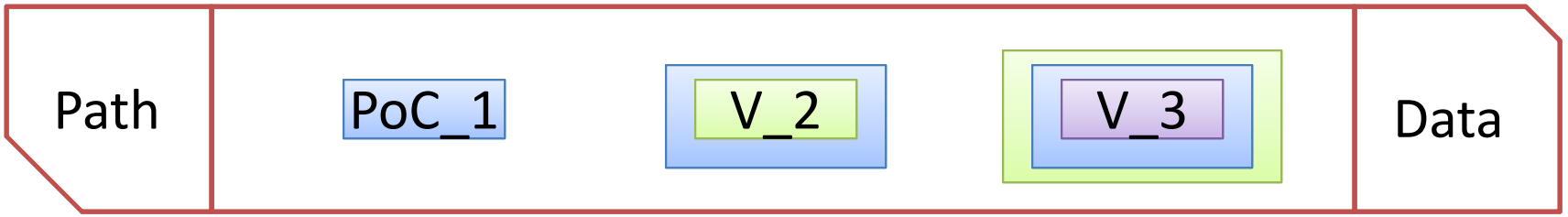


1. If not in cache, calculate $k_{2,3}$ = DH-Key-Exch(R1, R2)
2. Set V_3 = V_3 ^ MAC($k_{2,3}$, 1 || Hash(Path || Data))

Challenge:
Enforcing policy decisions at line-rate

Step 2: Make sure the path is followed

1. Verify consent & provenance
2. Prove provenance

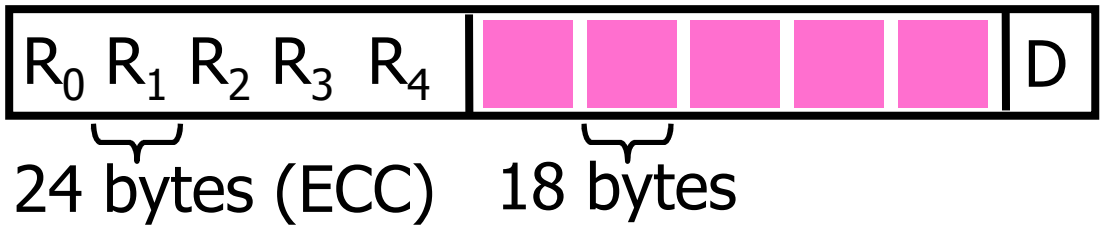


ICING's data plane in a nutshell

- Binds a packet to its path
  - Packet carries path (list of public keys), verifiers
  - Realms use $k_{i,j}$ to transform verifiers
  - R_i verifies provenance through upstream realms R_j using $k_{j,i}$
  - R_i proves provenance to downstream realms R_j using $k_{i,j}$
- No key distribution: R_i derives $k_{i,j}$ from R_j's name
- Resists attack: forgery, injection, short-circuiting, …
- Feasibility: is required space, computation tolerable?

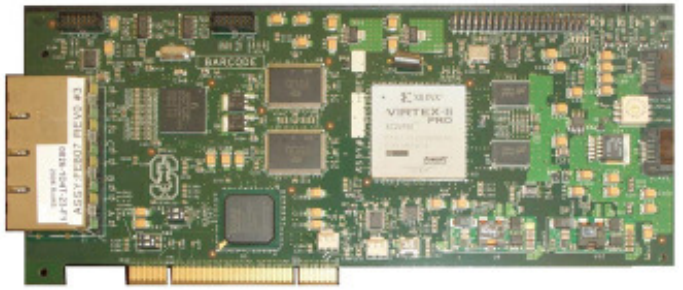
ICING is feasible

Space overhead?



- Average ICING header: ~250 bytes
- Average packet size: ~1300 bytes [CAIDA]
- So, total overhead from ICING: ~20% more space

ICING is feasible



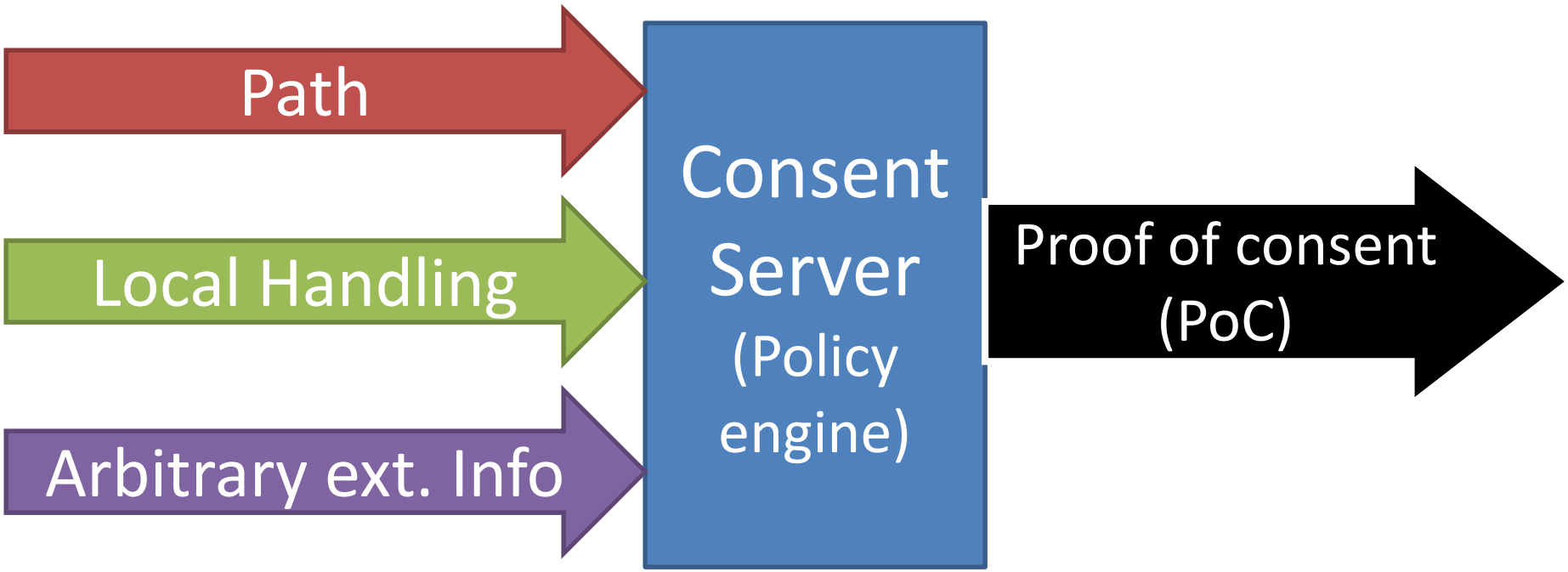
- What is the hardware cost?
  - NetFPGA gate counts: ICING is 13.4 M, IP is 8.7 M
  - NetFPGA forwarding speed: ICING is ~80% of IP
  - ICING vs. simple IP in gates/(Gbits/sec): ~2x
- Bandwidth and computation increasing faster than crypto costs

Outline

- How general is general?
  (What is the control? Who gets control? How can it be used?)
- How do we enforce policy decisions in the data plane?
- What is the control/data plane interface and how can it be used?

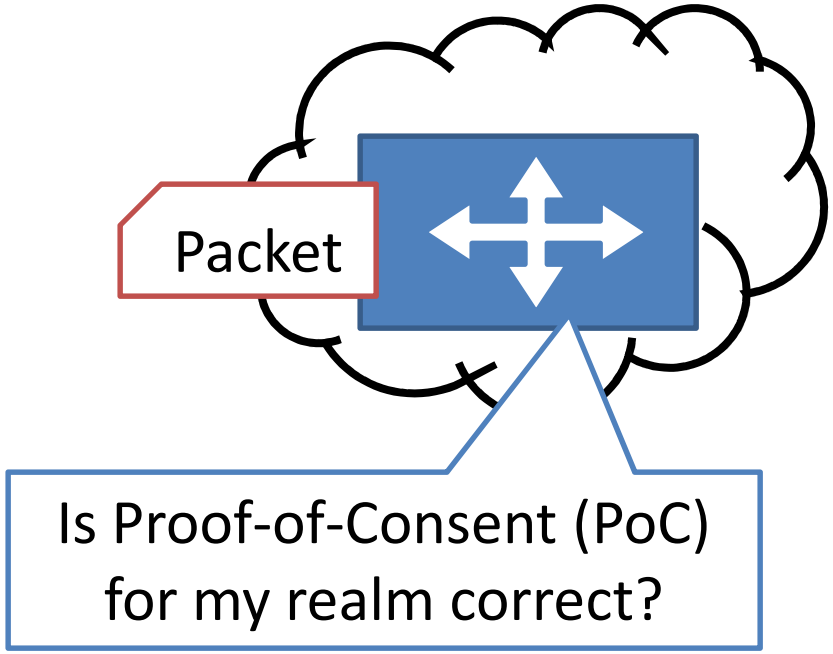
Control/Data Plane Interface

1. Allow/Deny Decisions



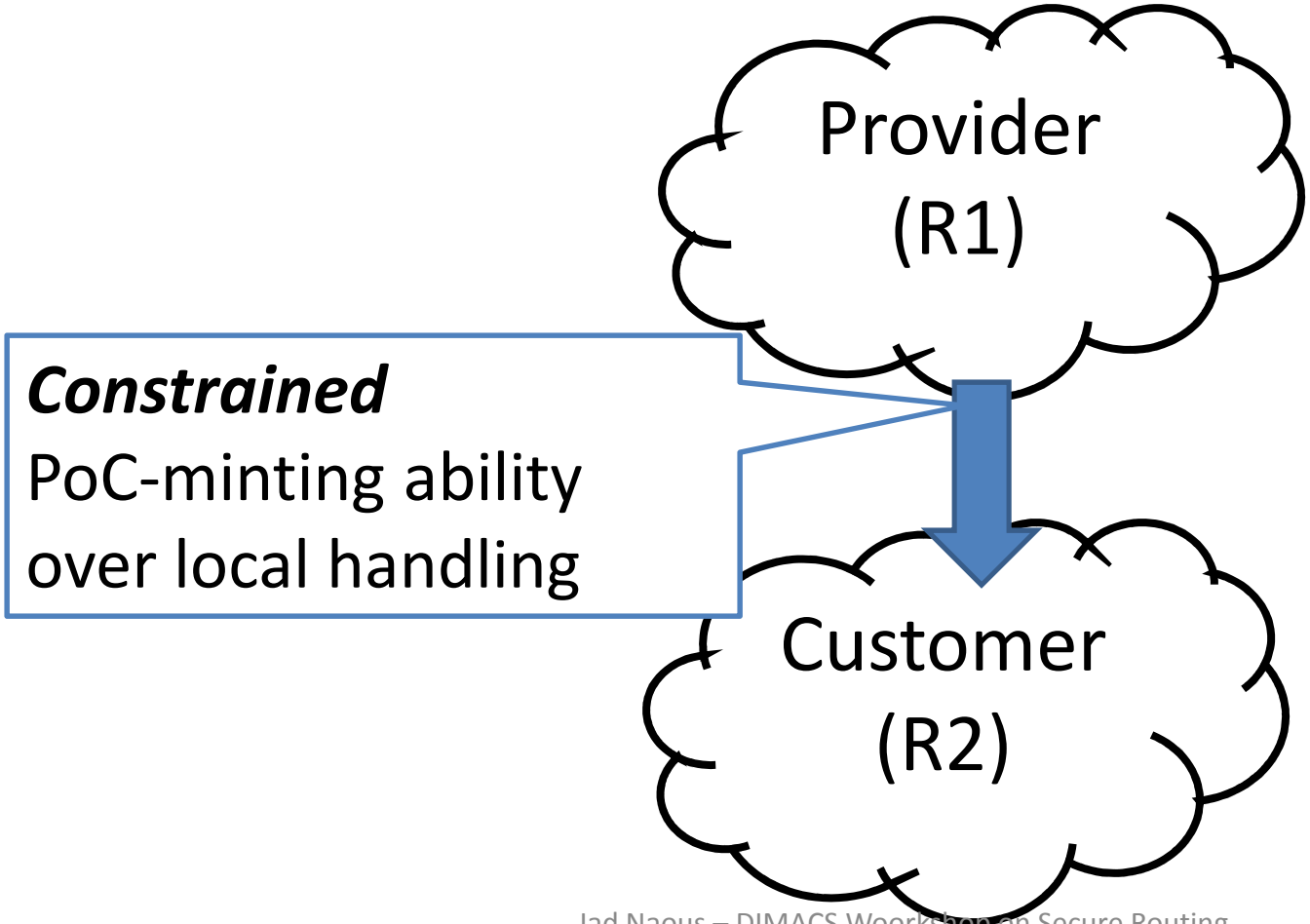
Control/Data Plane Interface

1. Allow/Deny Decisions



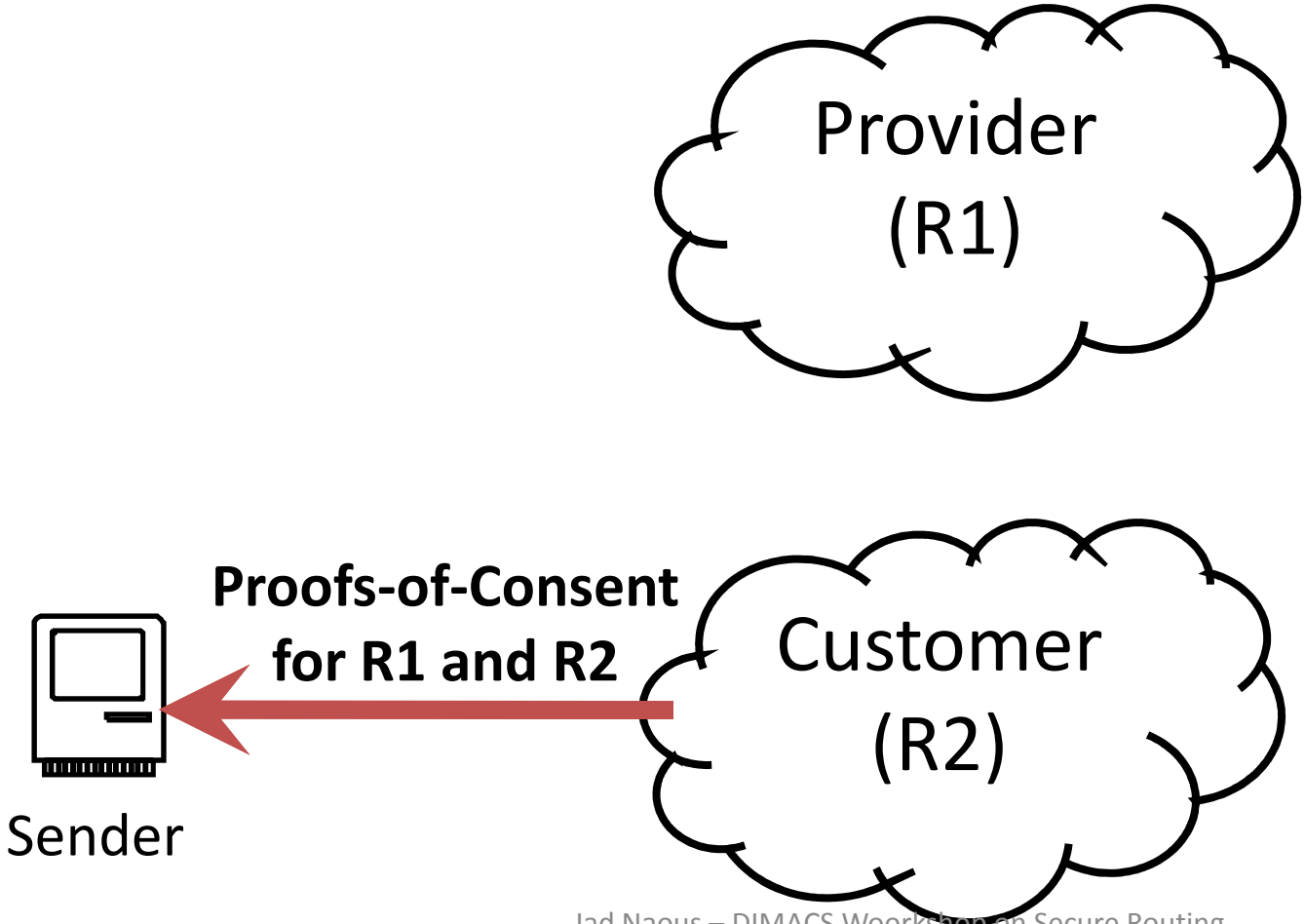
Control/Data Plane Interface

2. Allow/Deny Decision Delegation



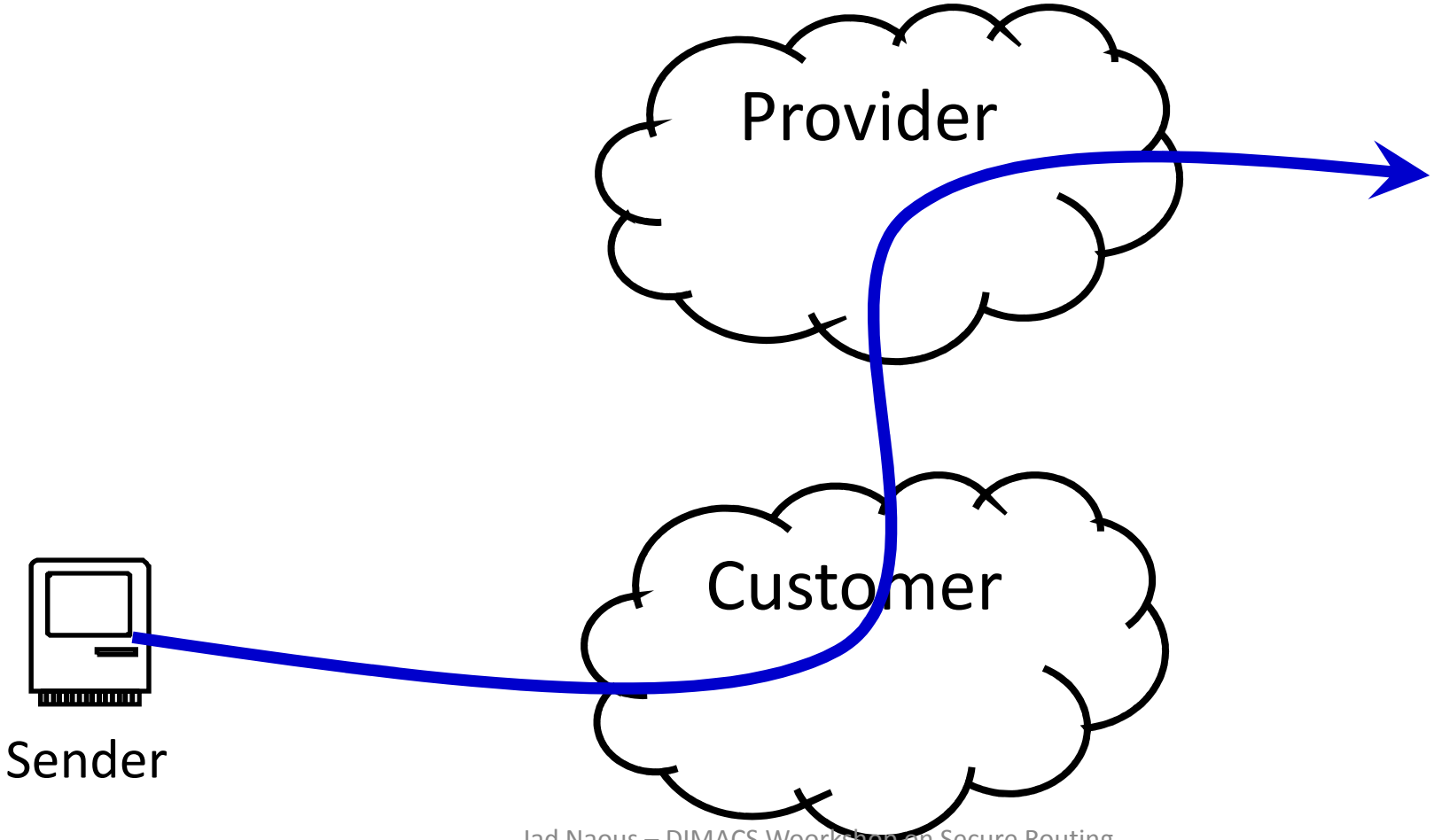
Control/Data Plane Interface

2. Allow/Deny Decision Delegation

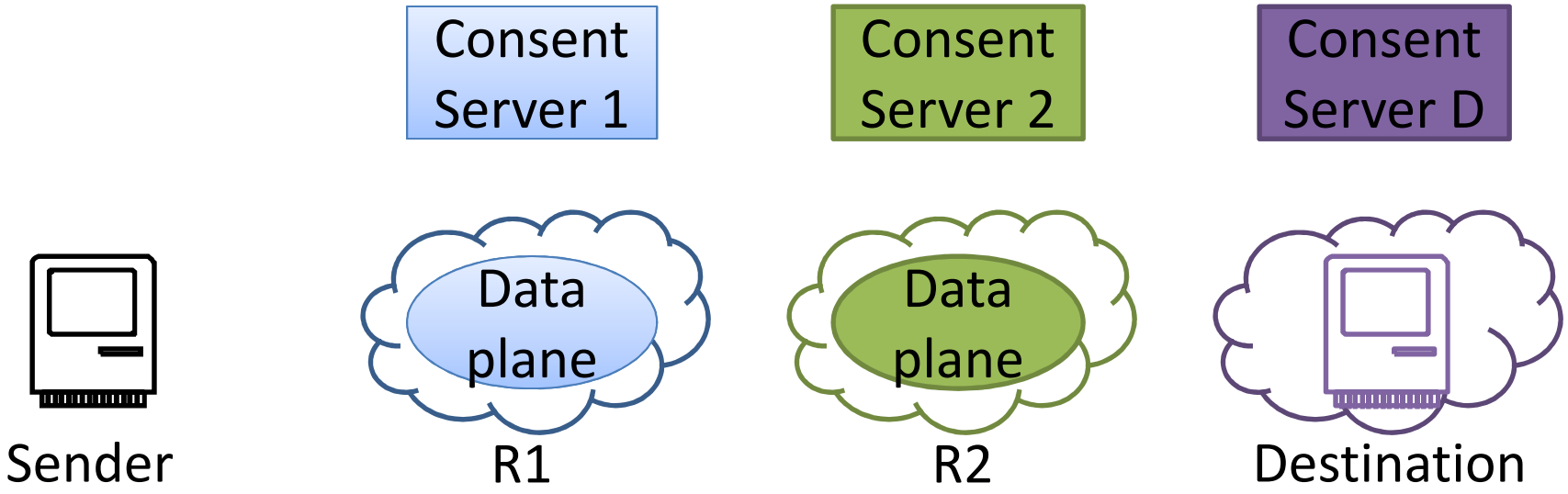


Control Plane “Knobs”

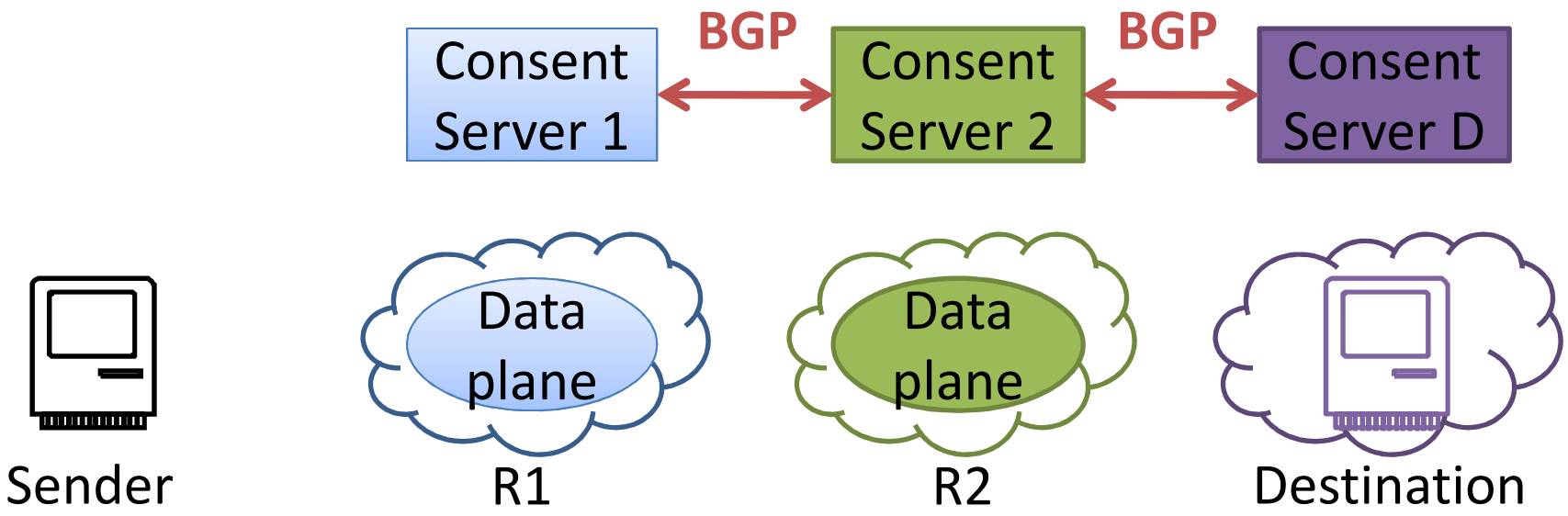
2. Allow/Deny Decision Delegation



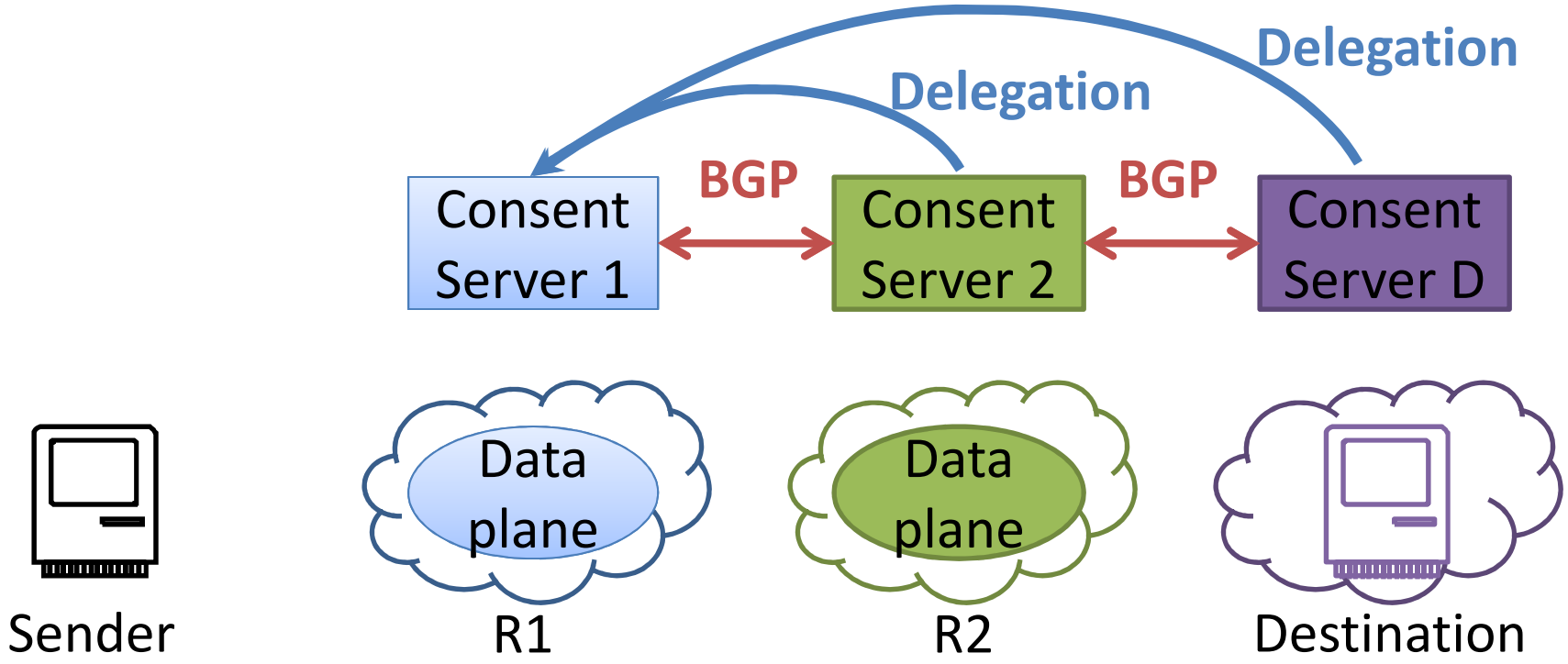
Example:
BGP with Enforcement



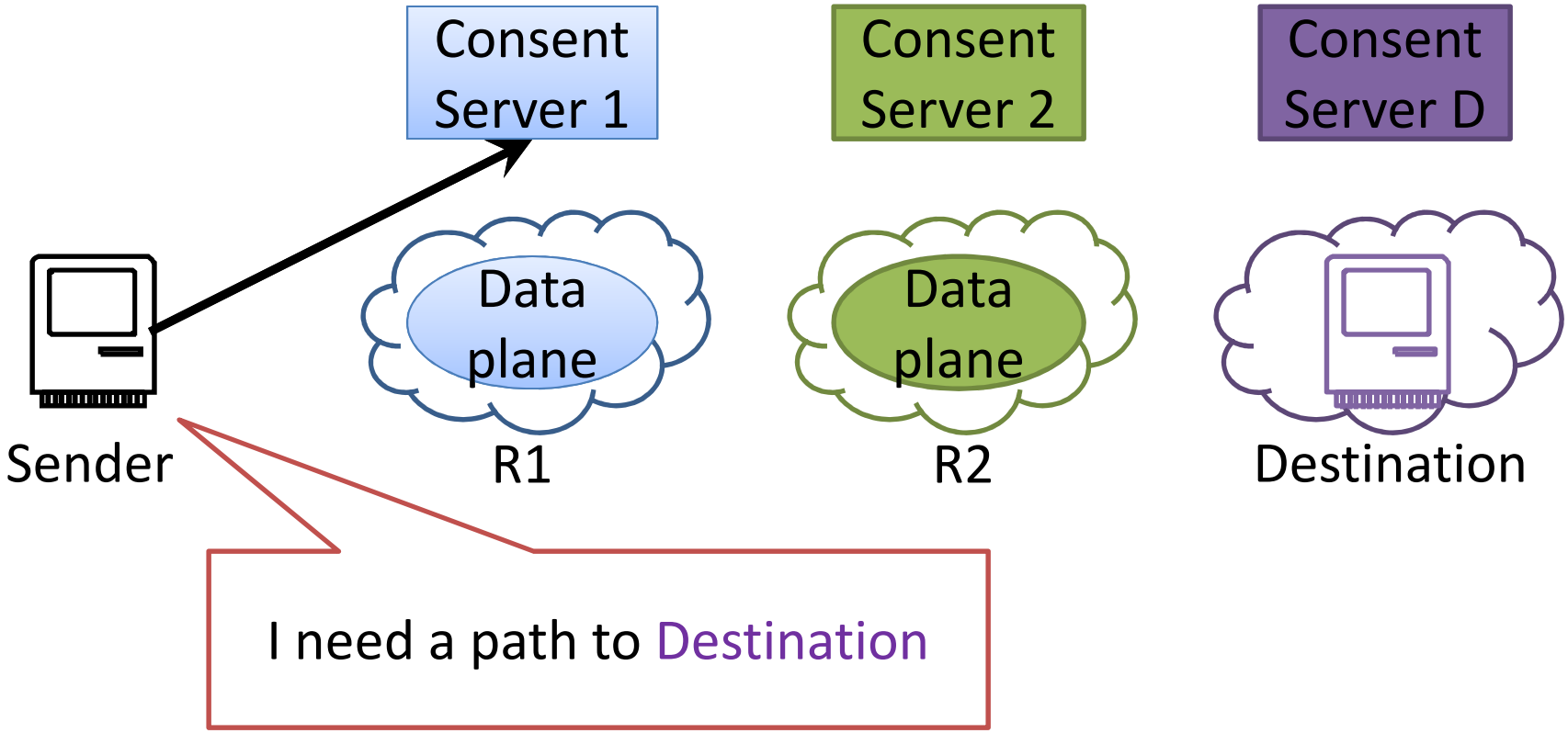
Example:
BGP with Enforcement



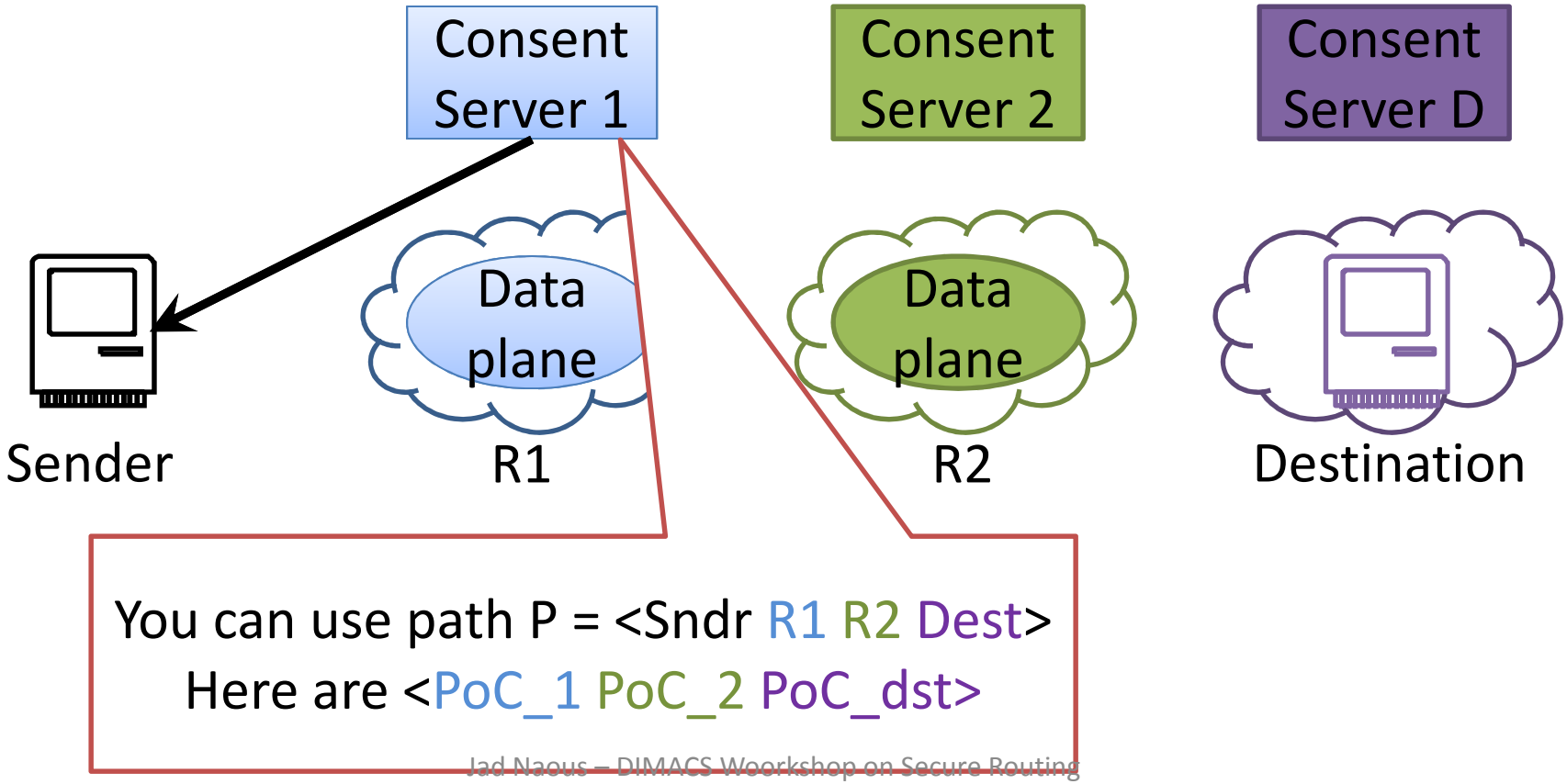
Example:
BGP with Enforcement



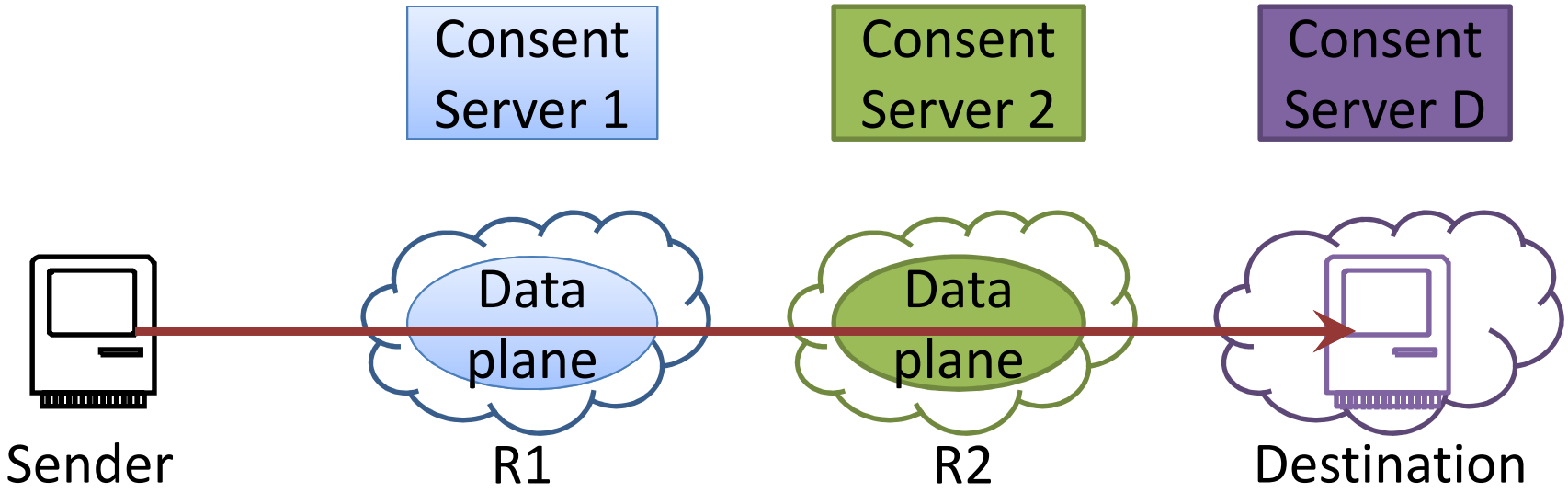
Example:
BGP with Enforcement



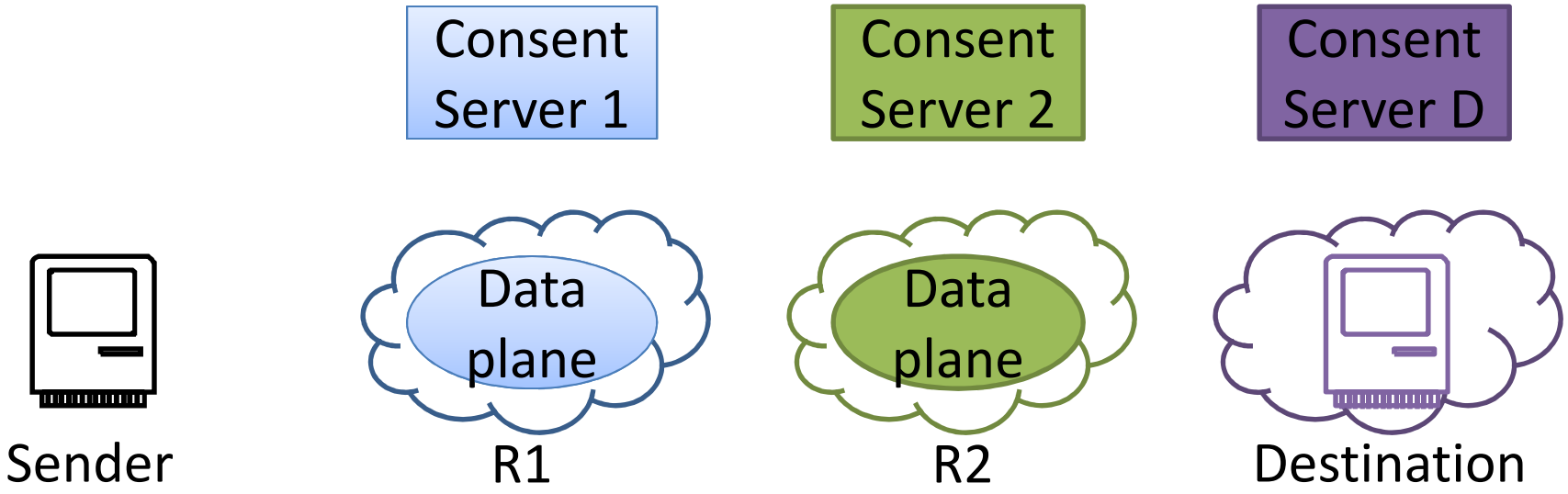
Example:
BGP with Enforcement



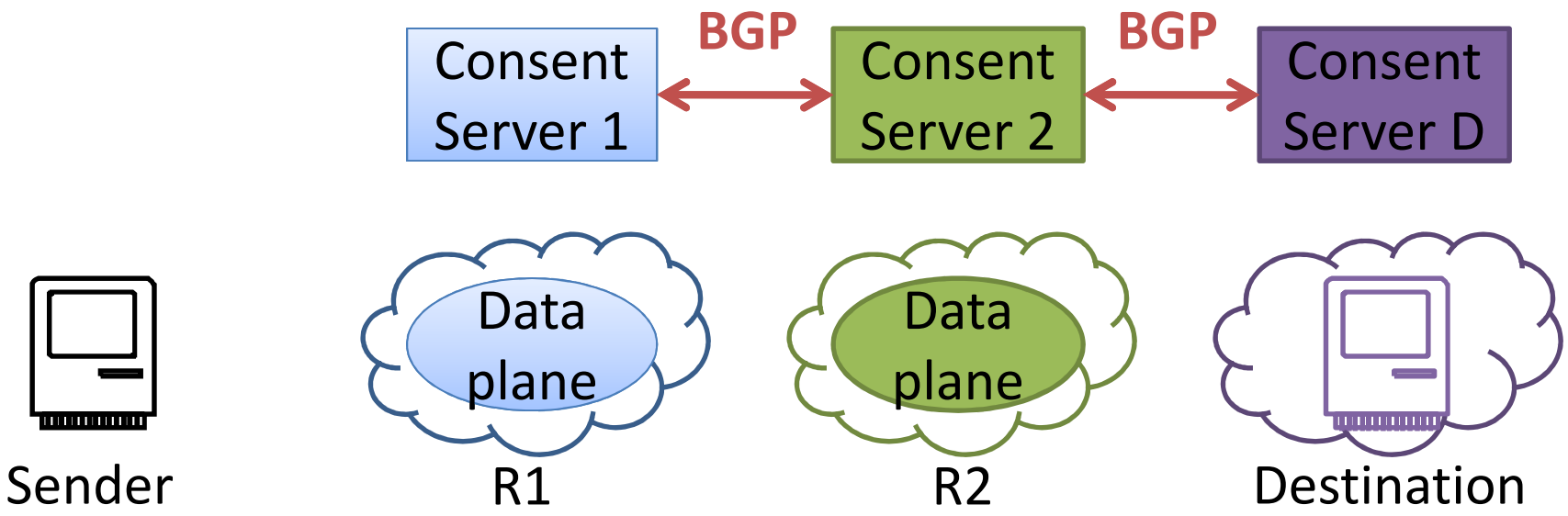
Example:
BGP with Enforcement



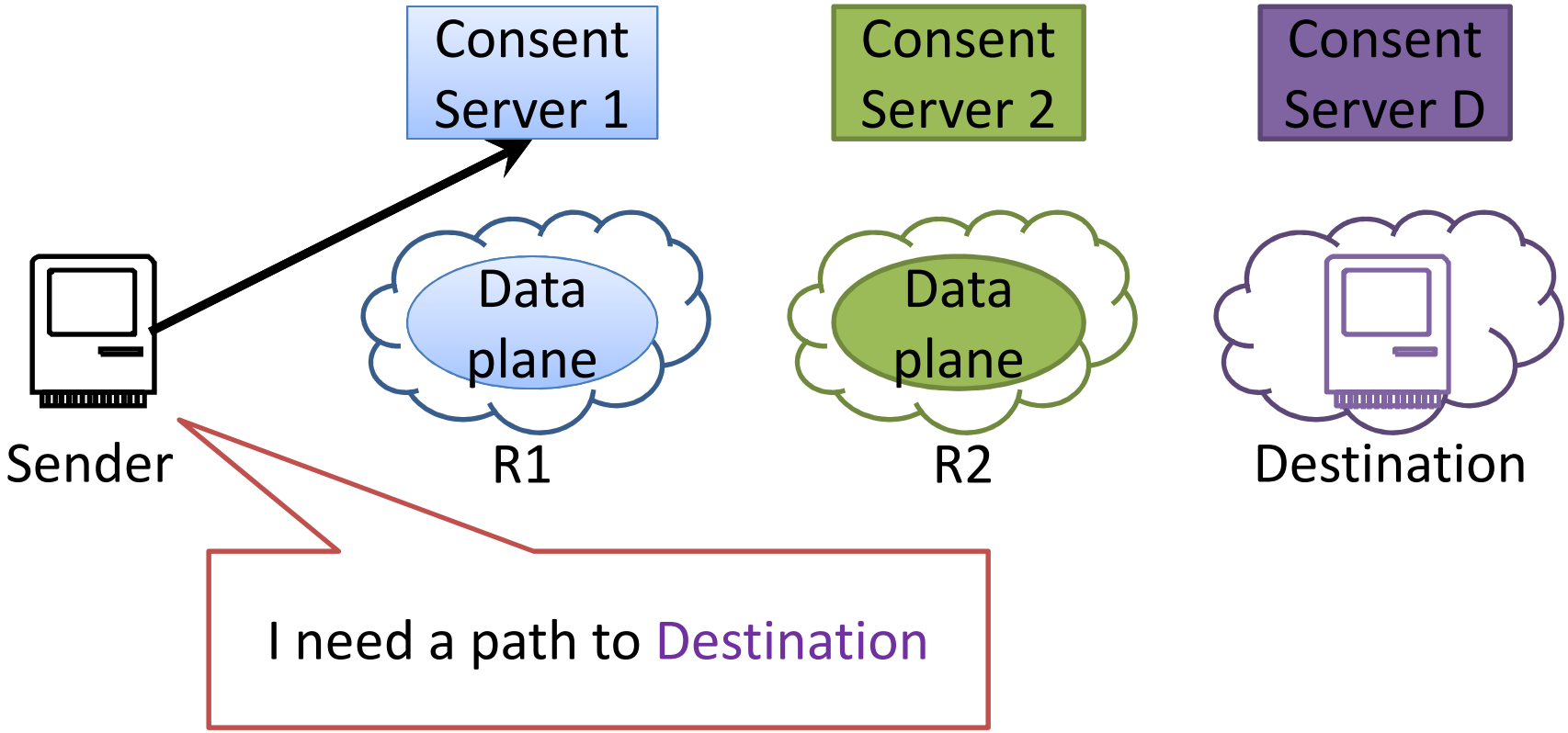
Example:
TVA and default-off



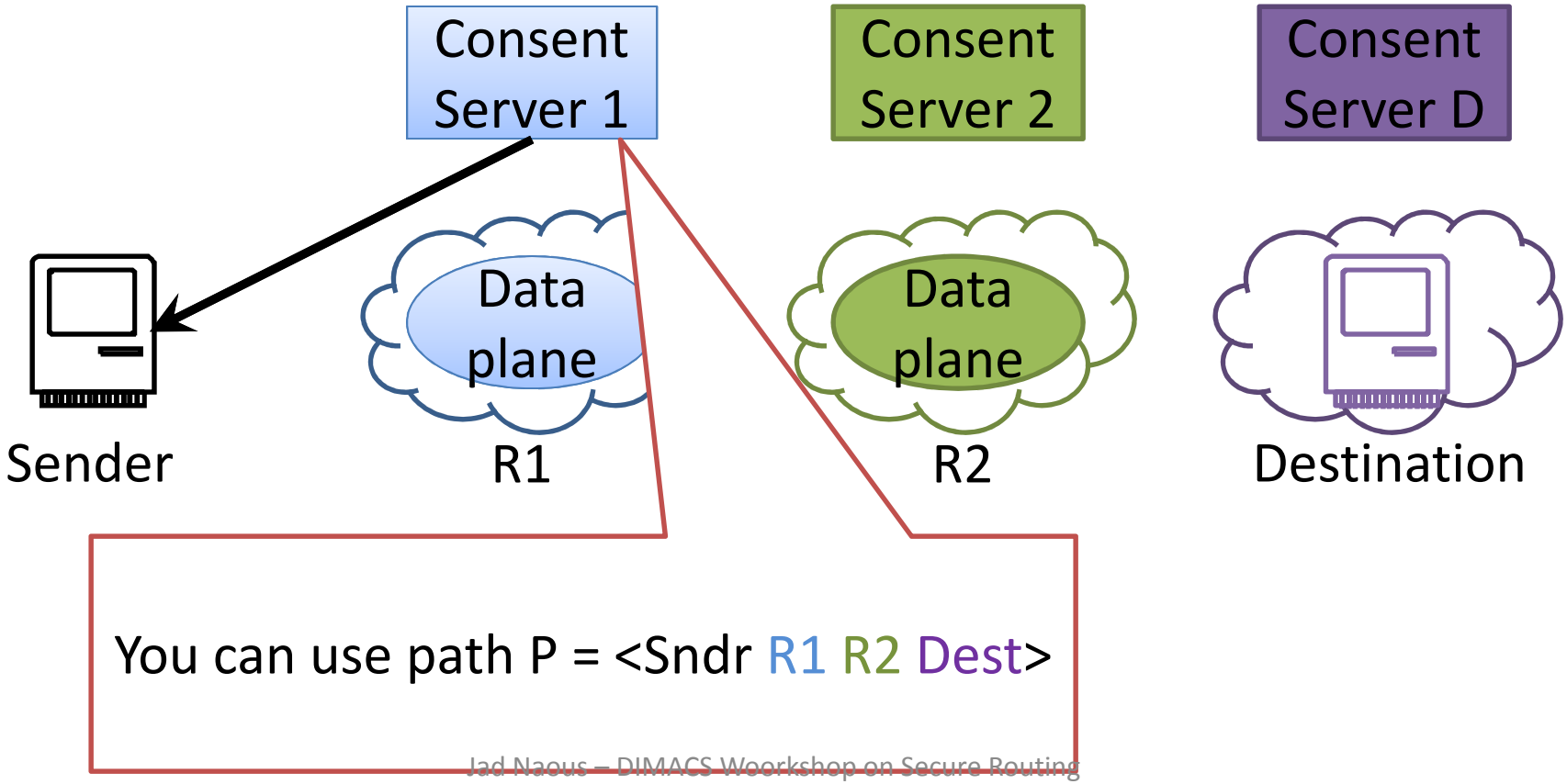
Example:
TVA and default-off



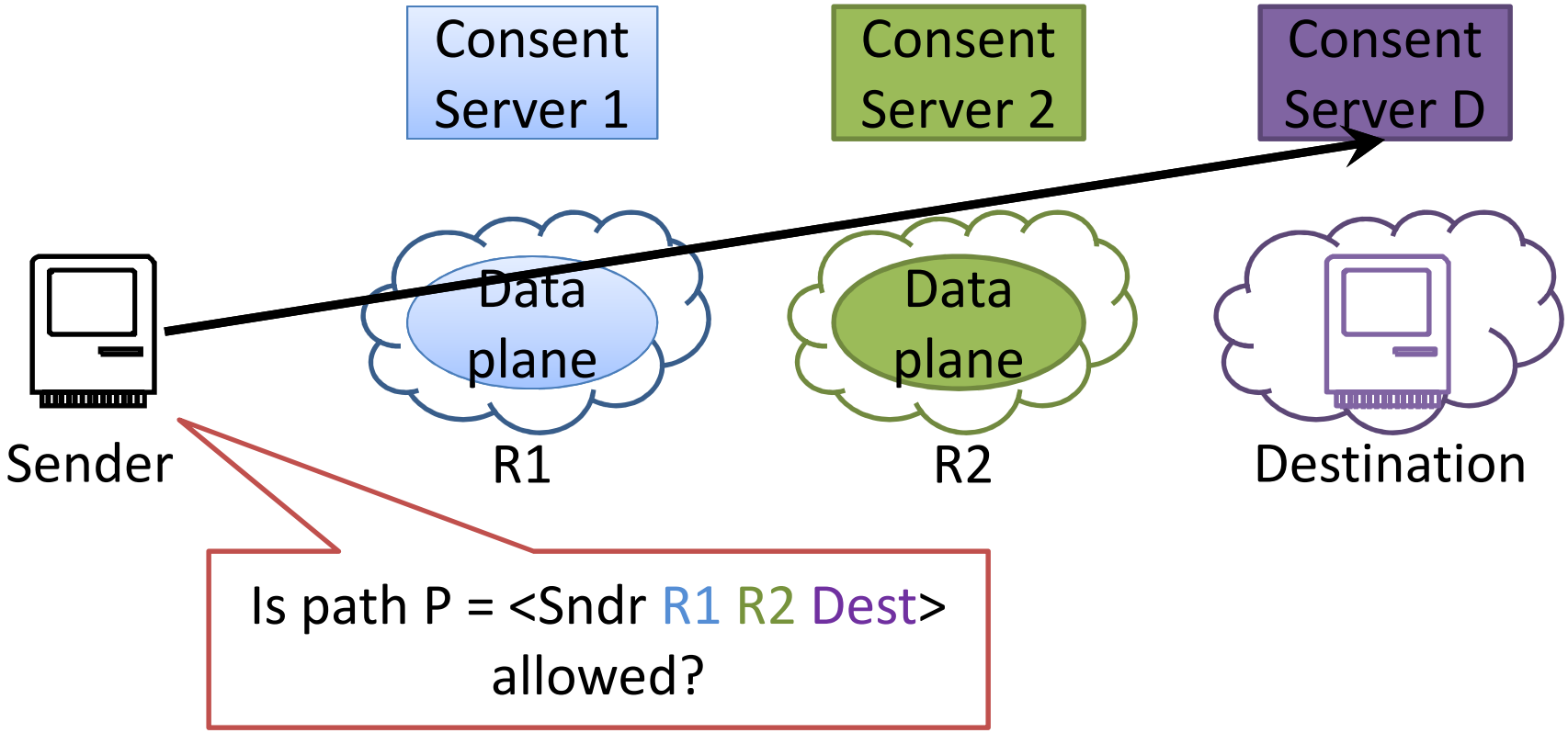
Example:
BGP with Enforcement



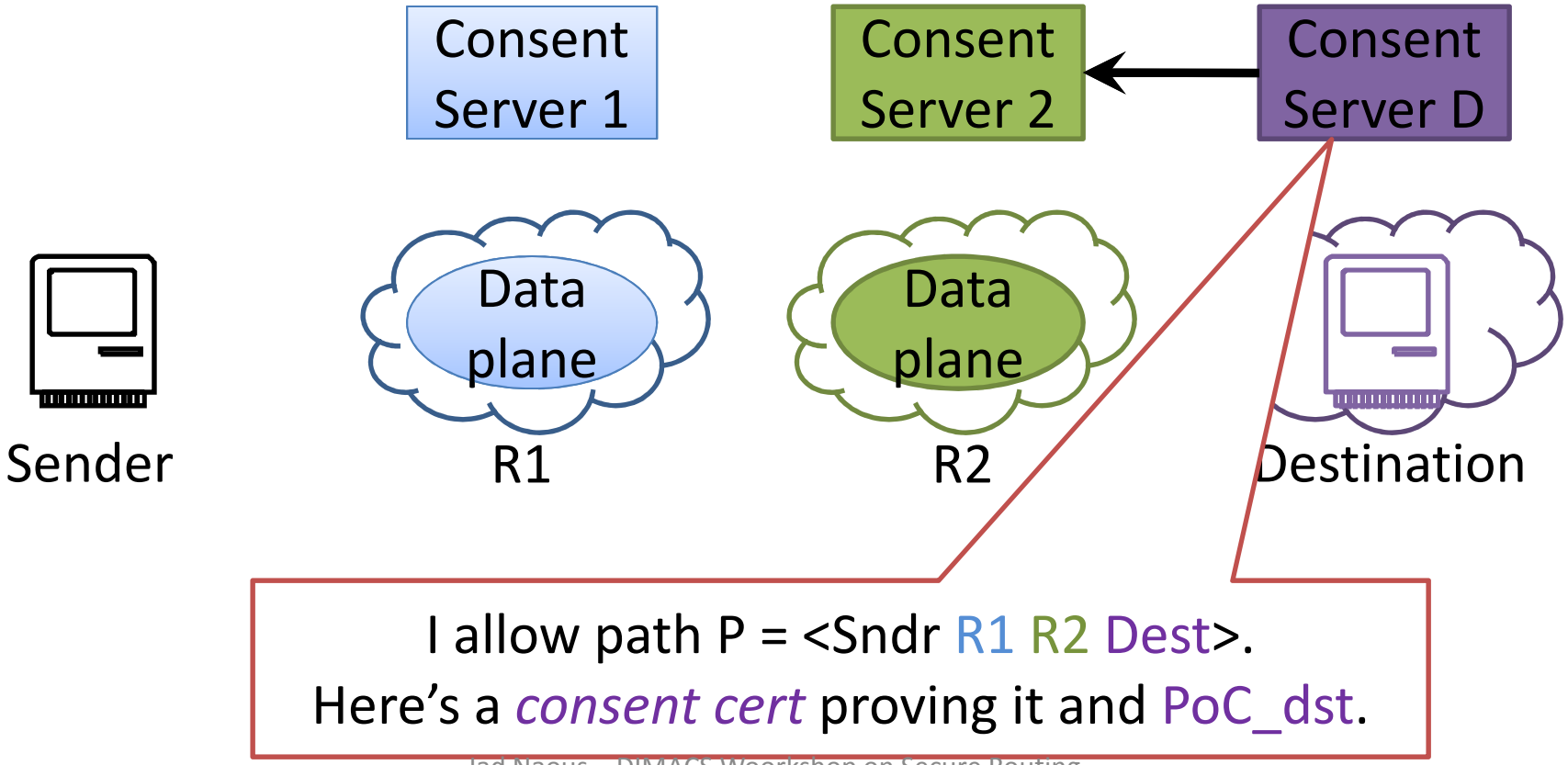
Example:
TVA and default-off



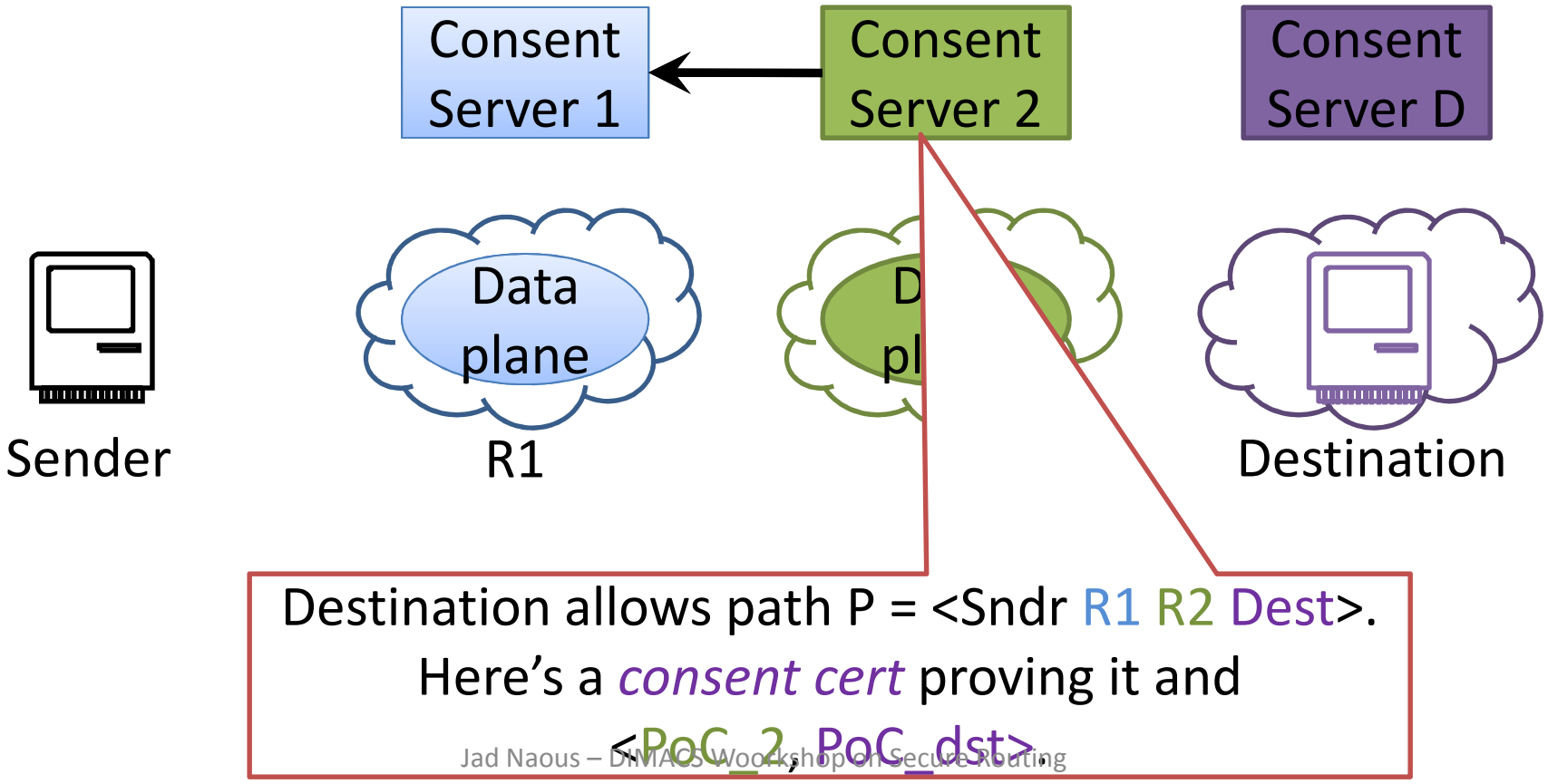
Example:
TVA and default-off



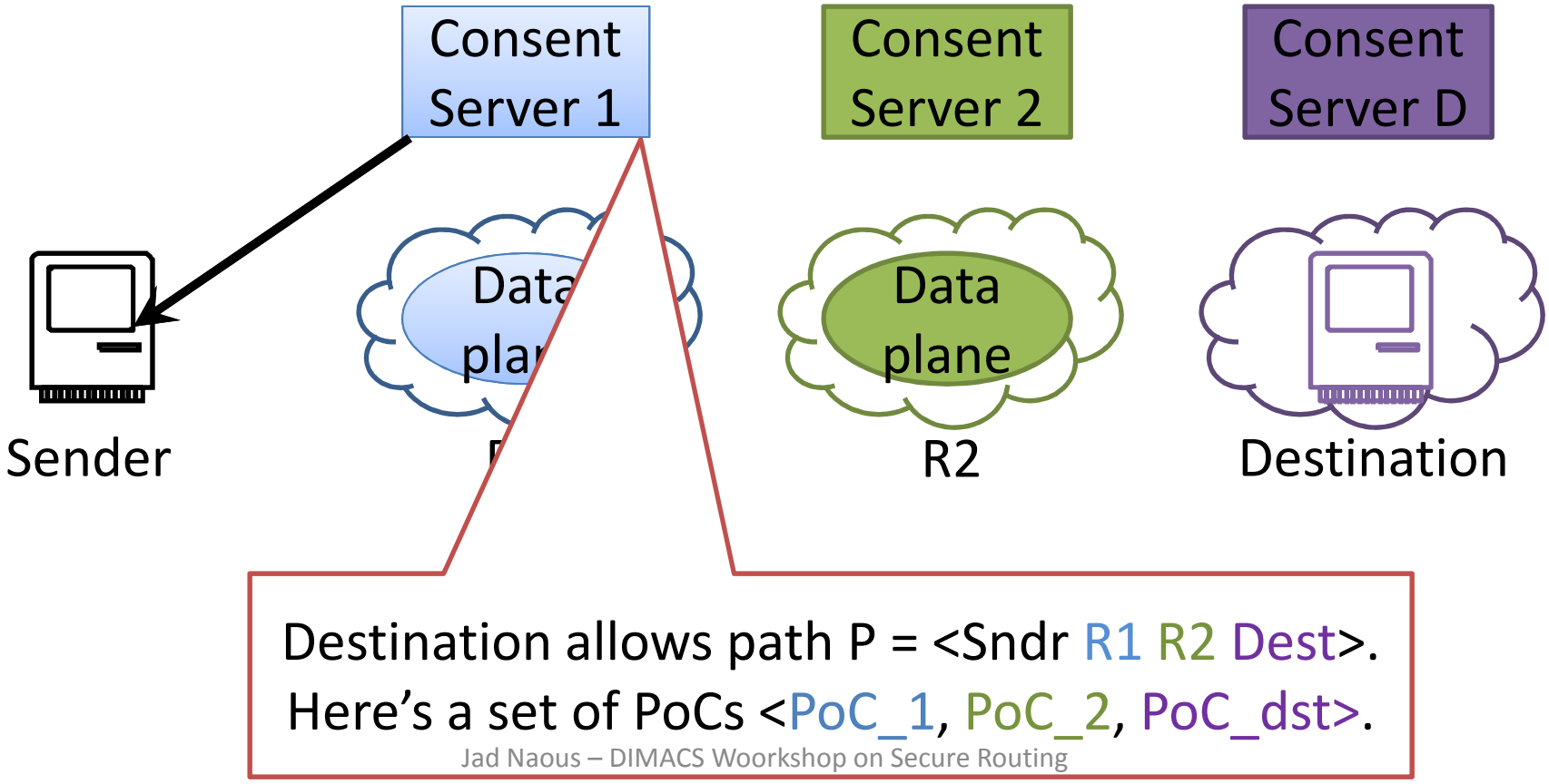
Example:
TVA and default-off



Example:
TVA and default-off



Example:
TVA and default-off

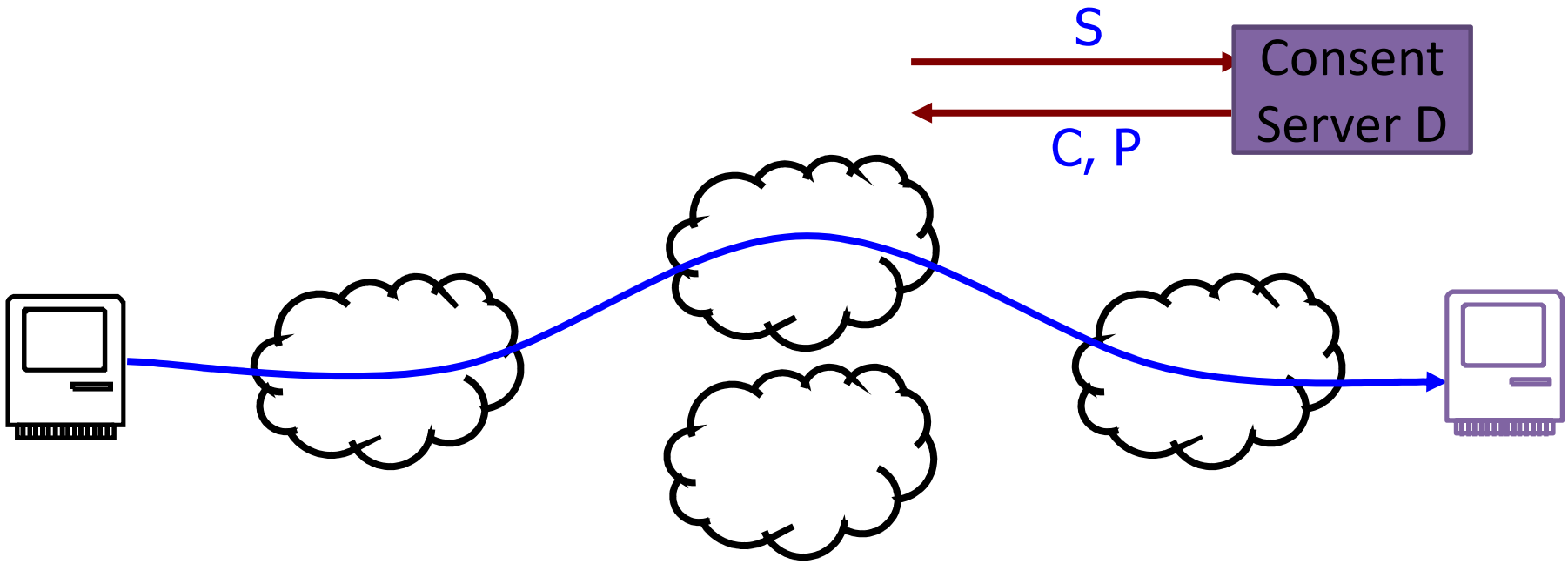


Others

Can emulate other proposals: NIRA, Pathlets, Source Routing, LSRR, …

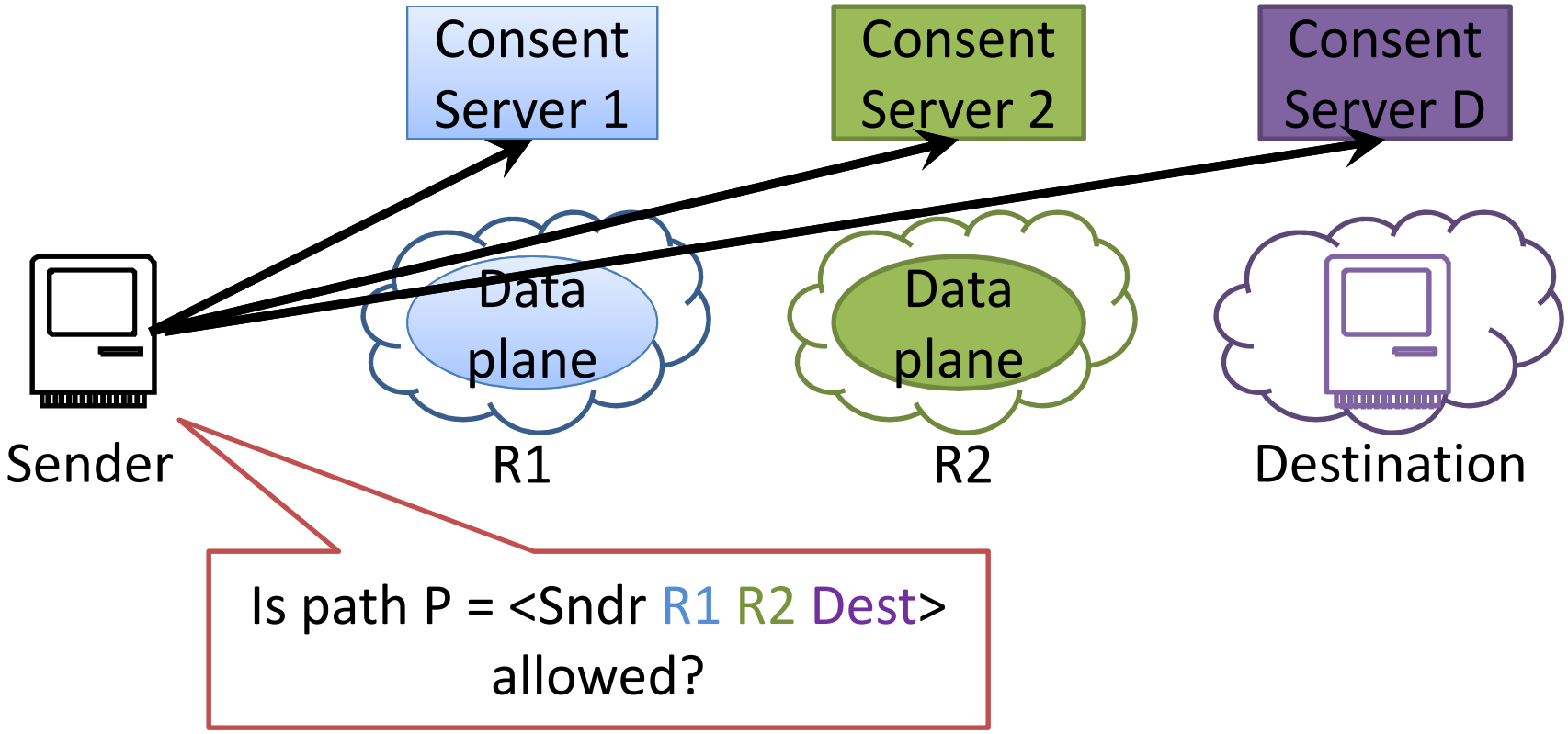
New policy engines with more features.

Example: choosing trustworthy providers through sink routing

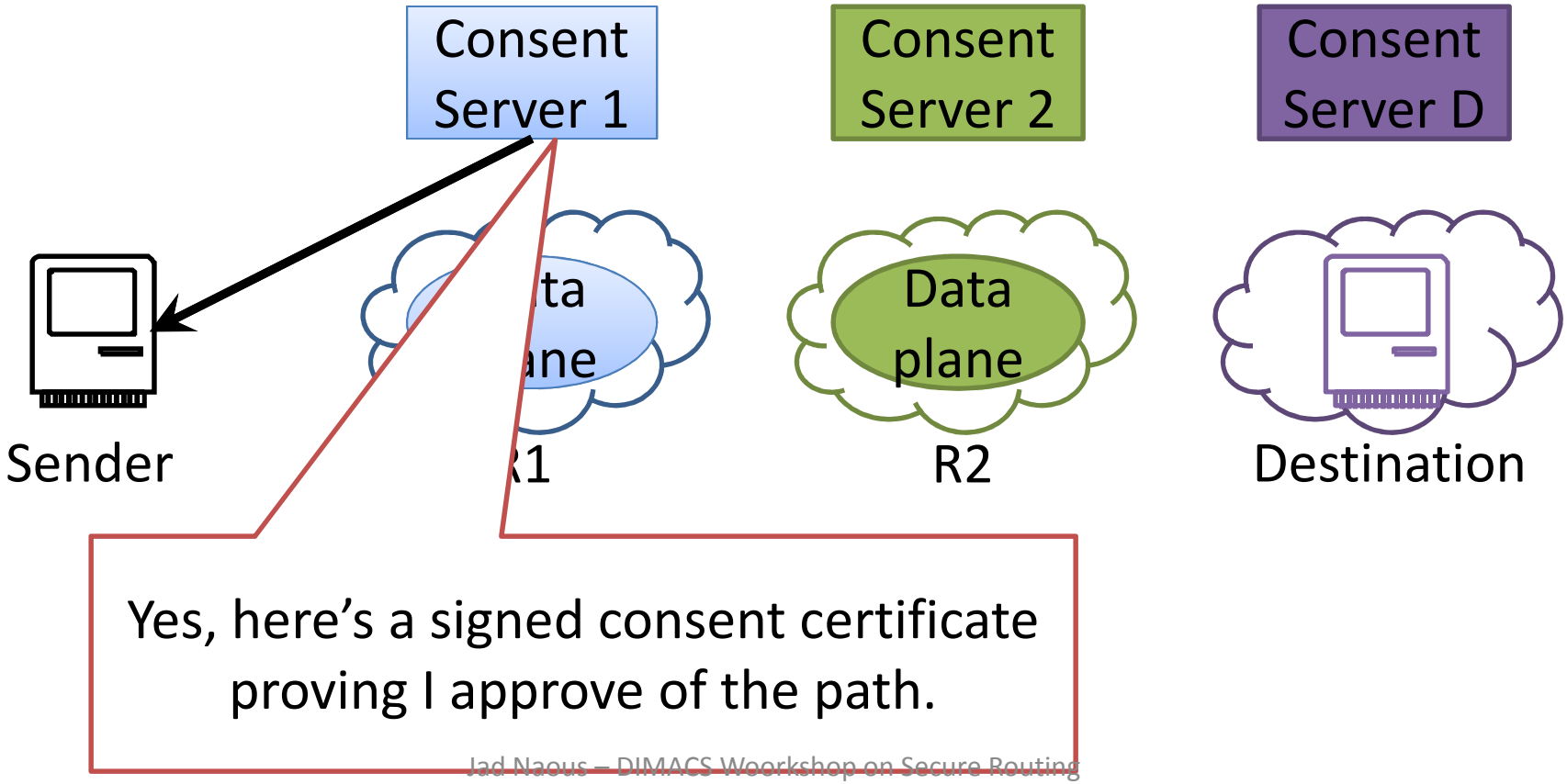


- This is analog of well-known source routing
- Sender requests consent; gives its own id (S)
- Receiver specifies path toward itself
  - Useful for organizations handling sensitive data

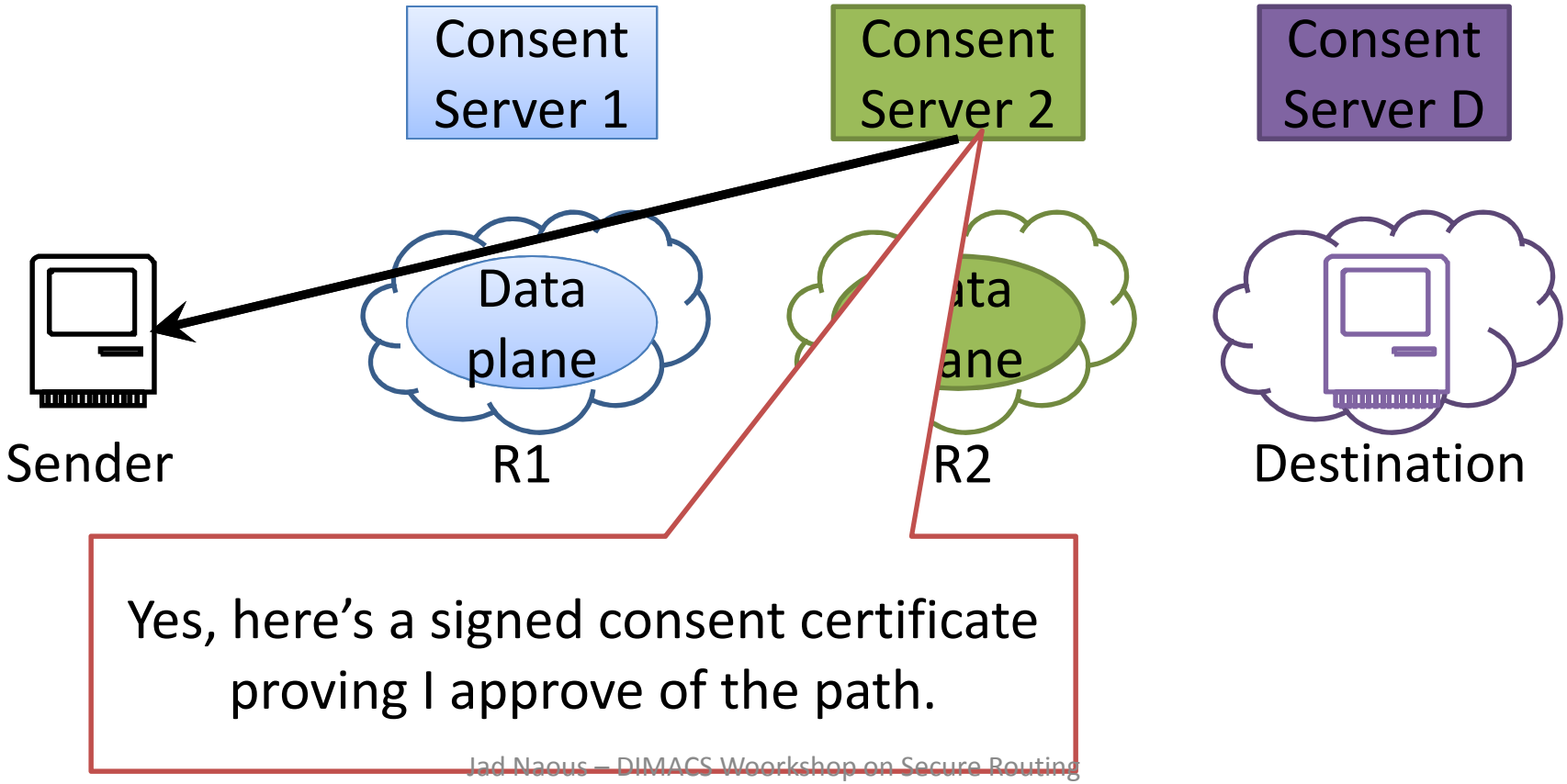
Example:
Early blocking of illegal packets



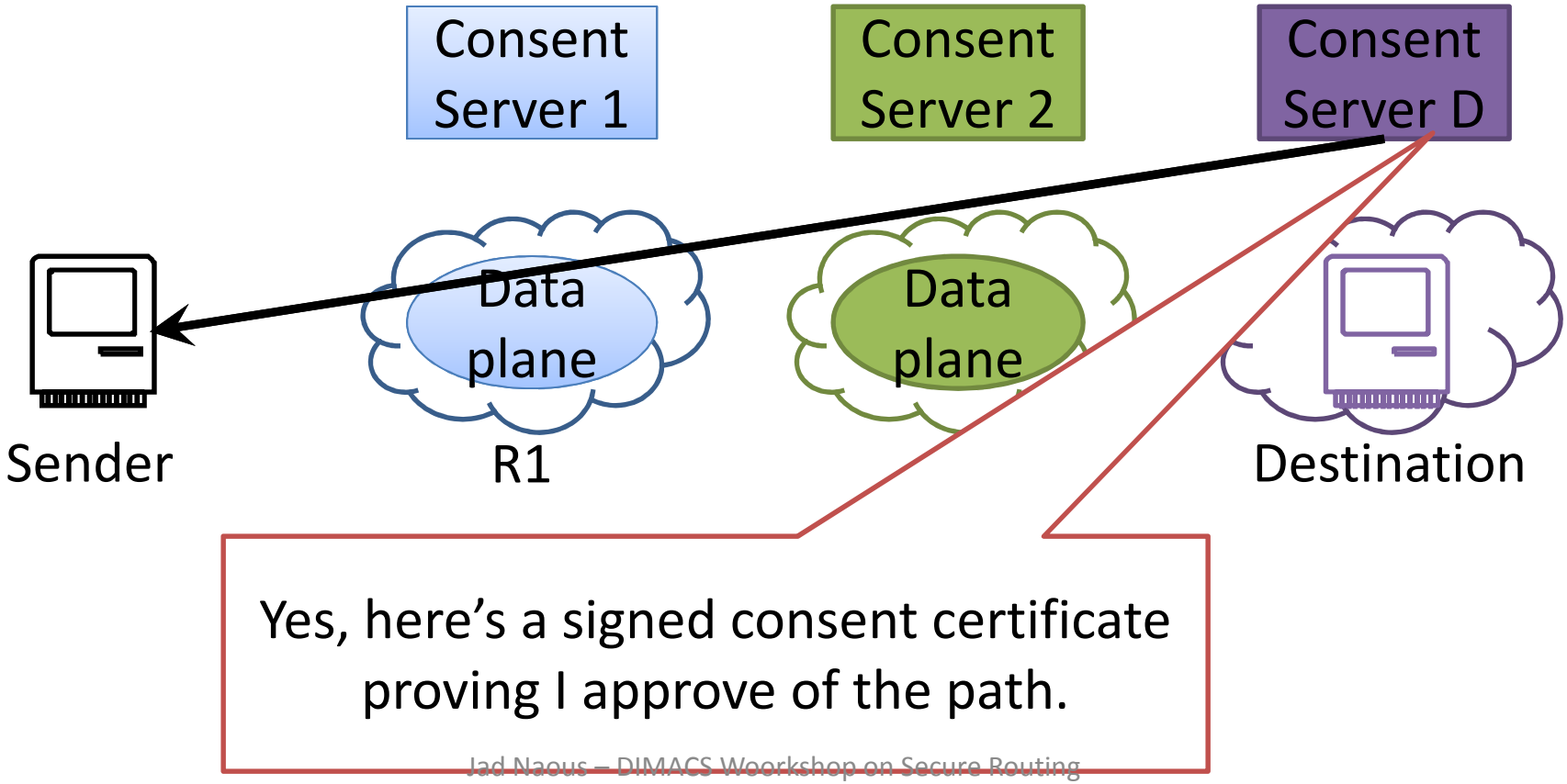
Example:
Early blocking of illegal packets



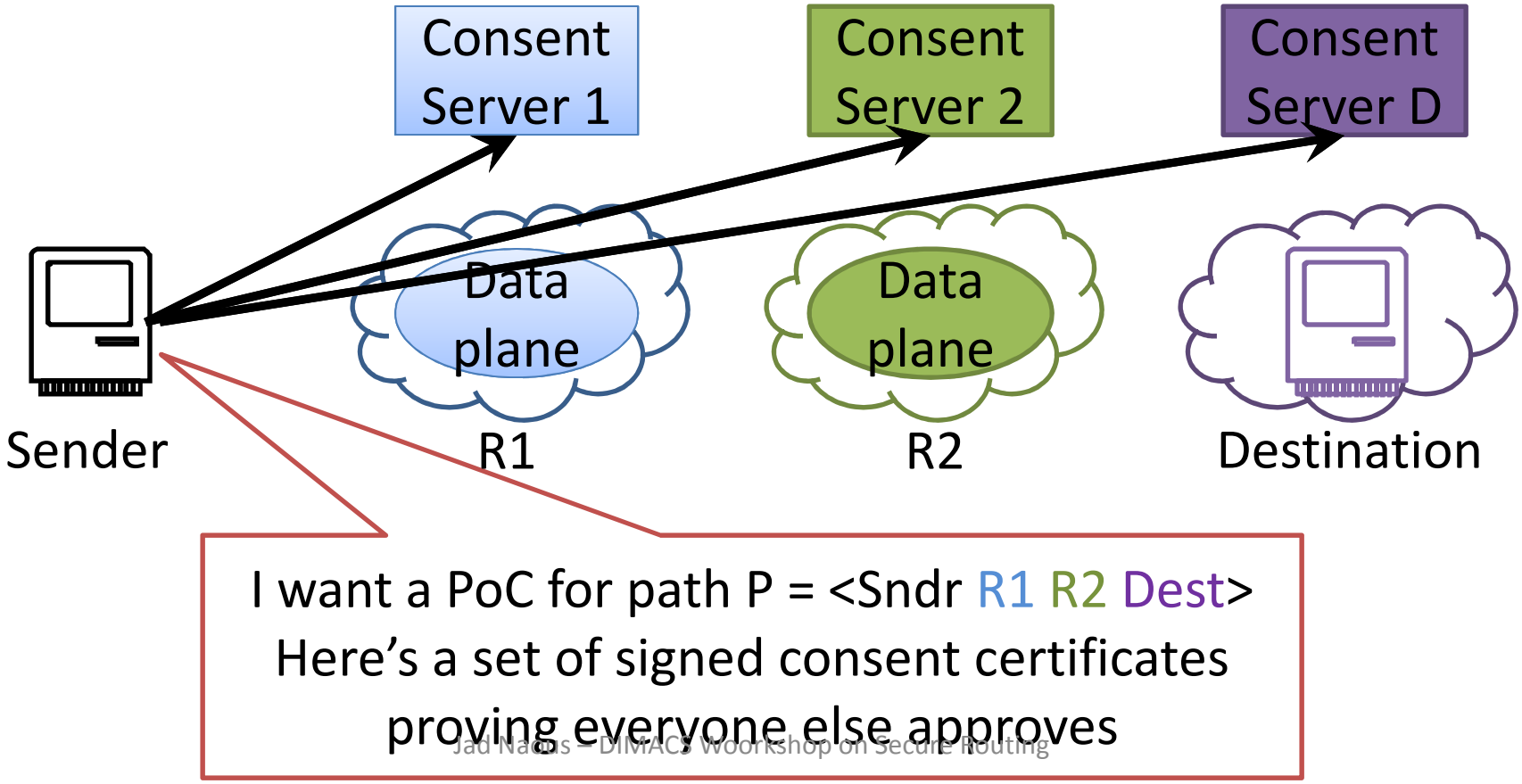
Example:
Early blocking of illegal packets



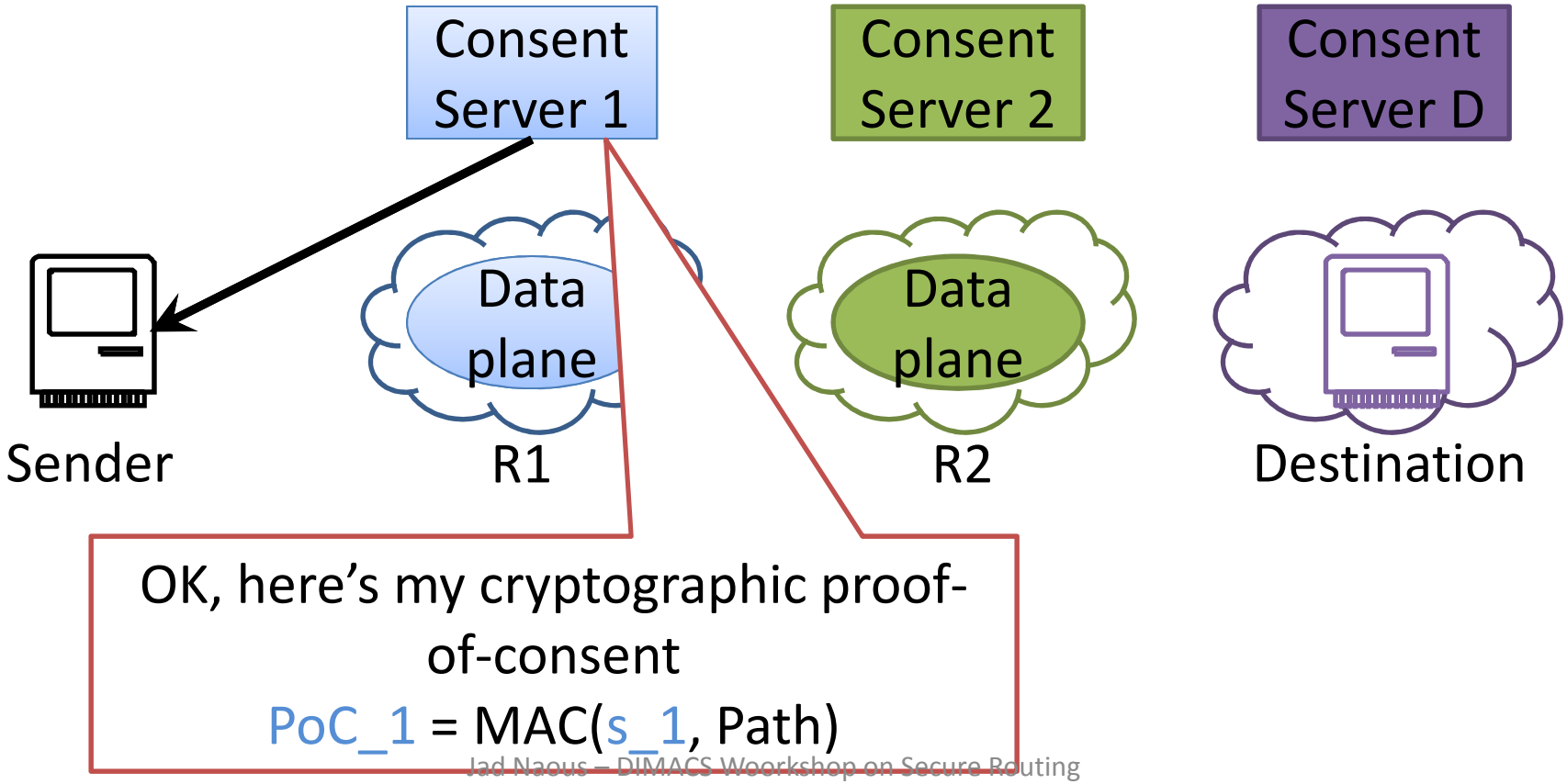
Example:
Early blocking of illegal packets



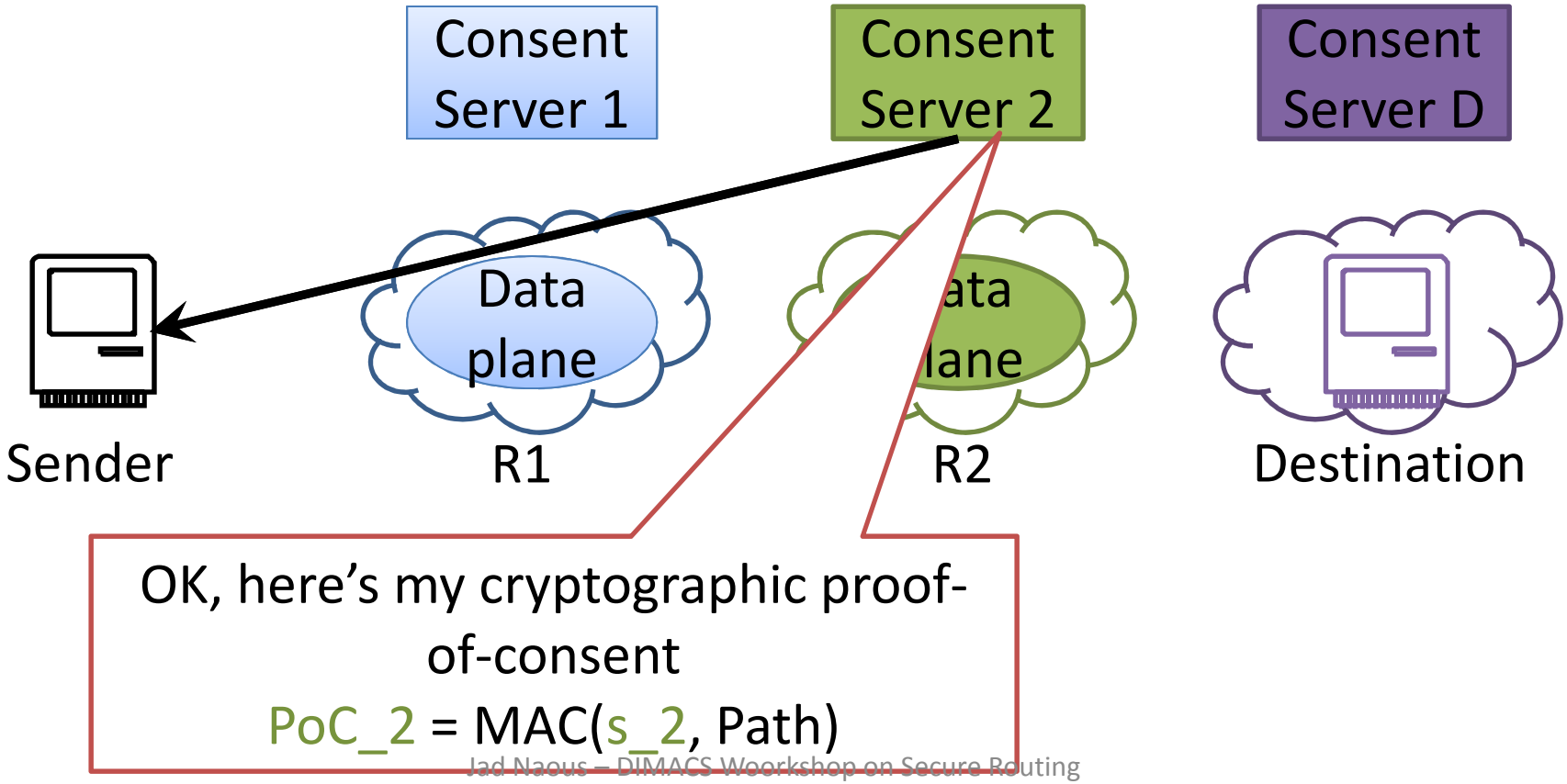
Example:
Early blocking of illegal packets



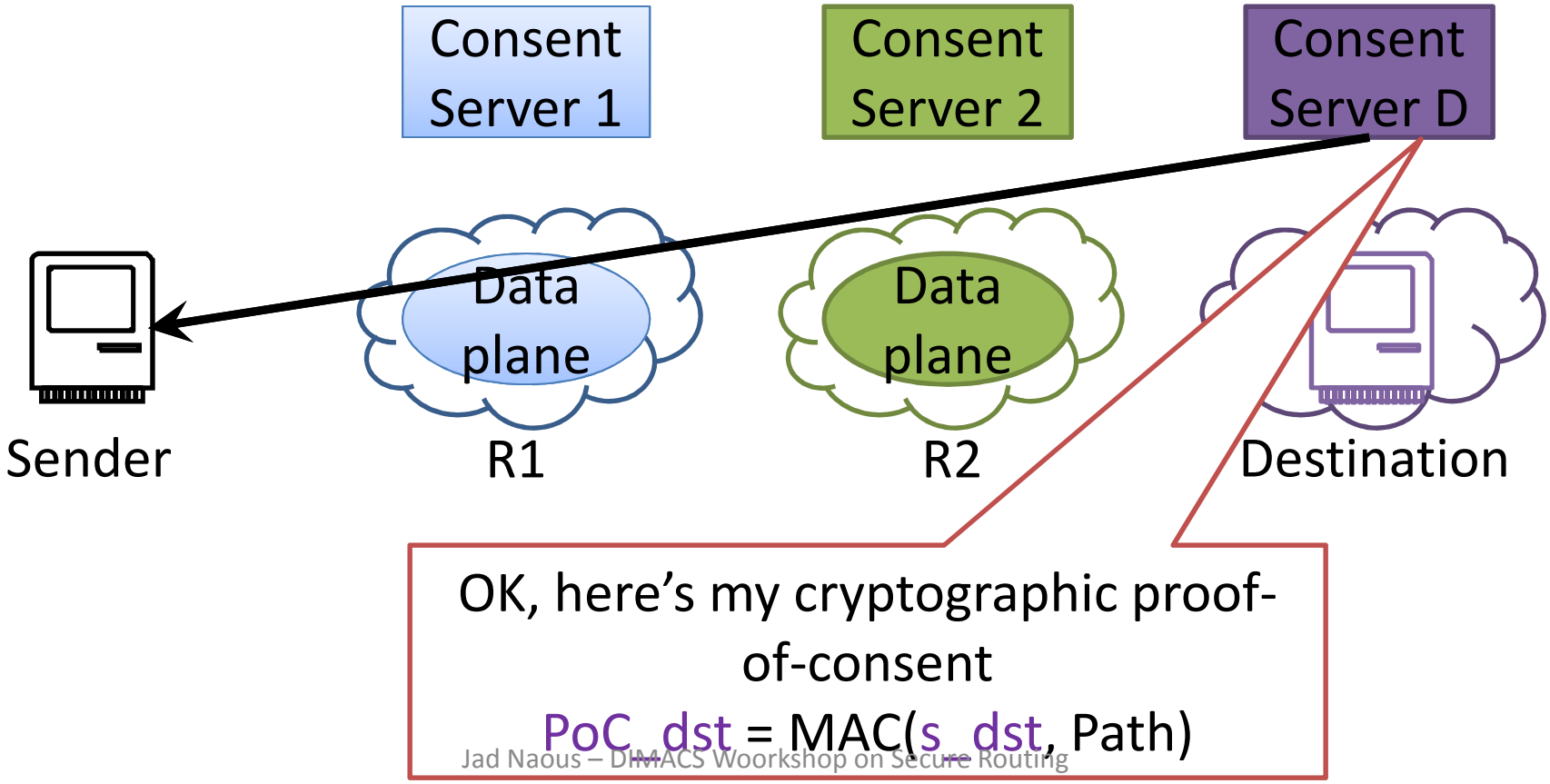
Example:
Early blocking of illegal packets



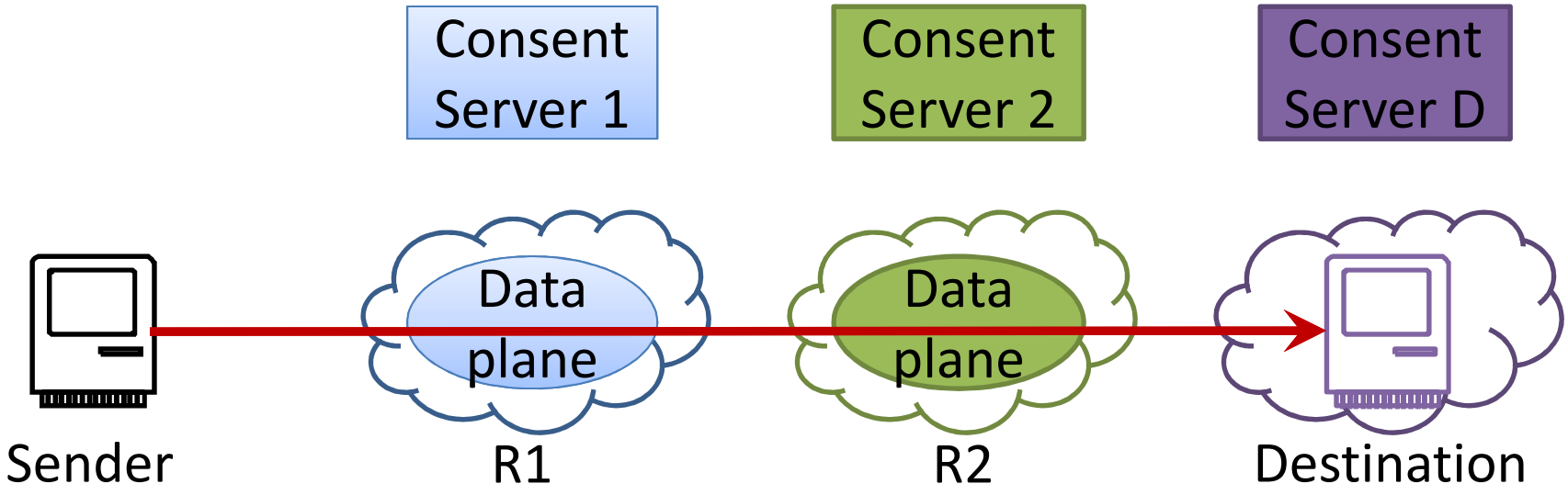
Example:
Early blocking of illegal packets



Example:
Early blocking of illegal packets

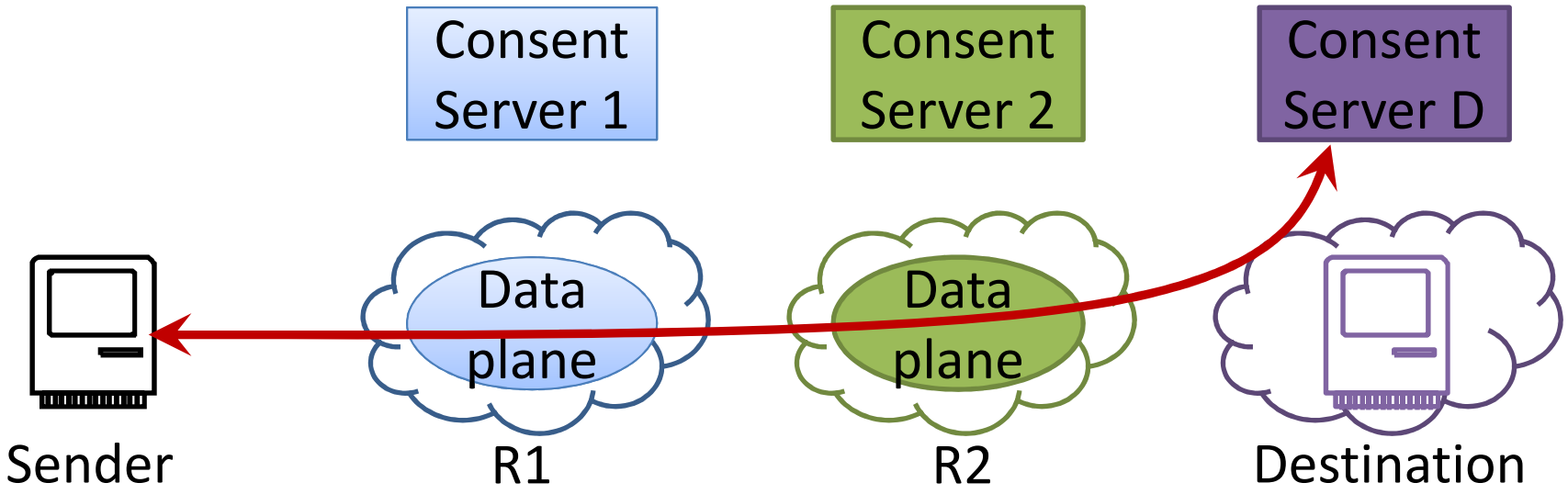


Example:
Early blocking of illegal packets

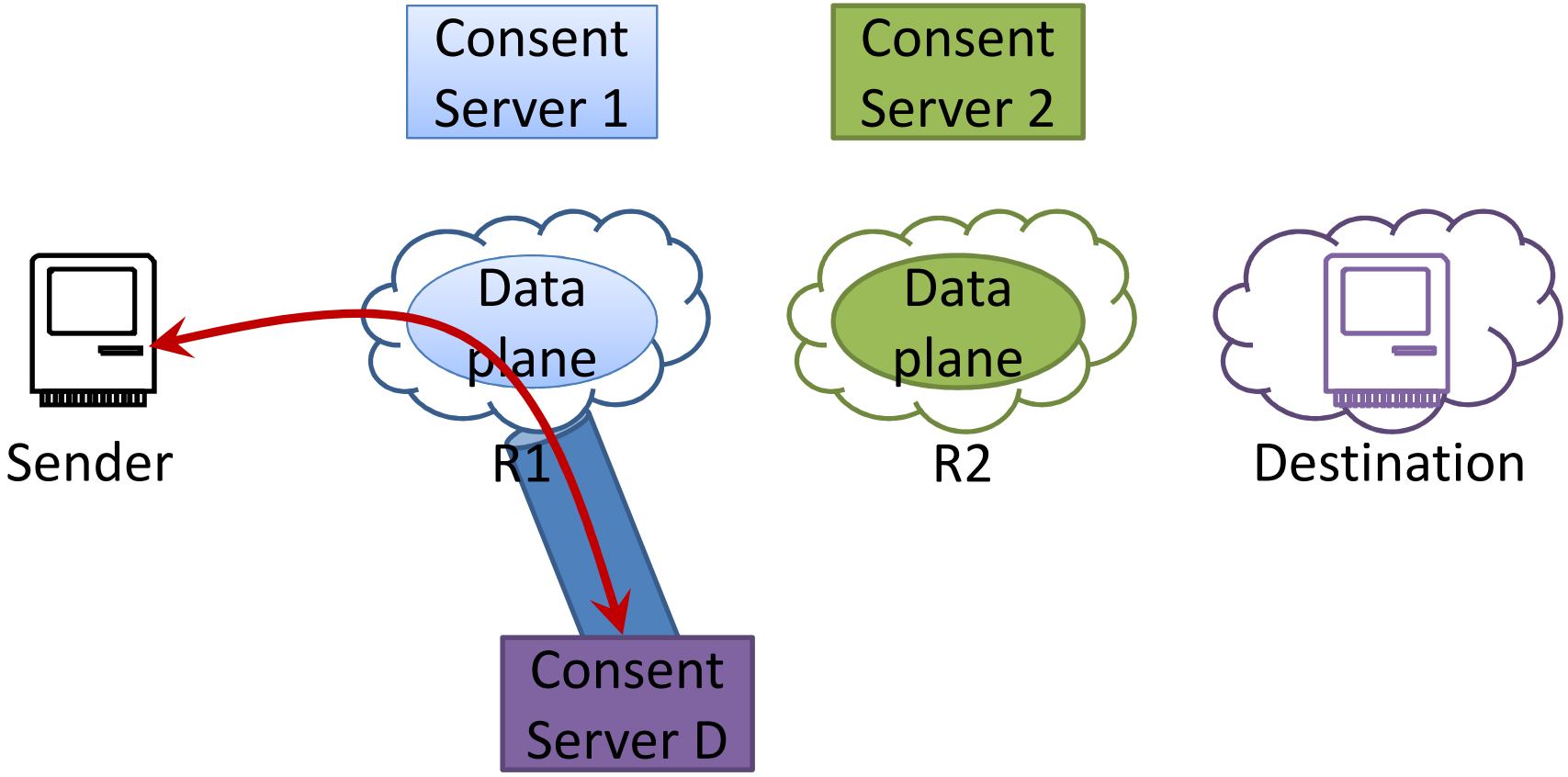


Packet =
<Path, PoC_1, PoC_2, PoC_dst, data>
PoCs verifiable by data plane using
Shared secret keys s_1, s_2, s_dst

Example use: preventing denial-of-service



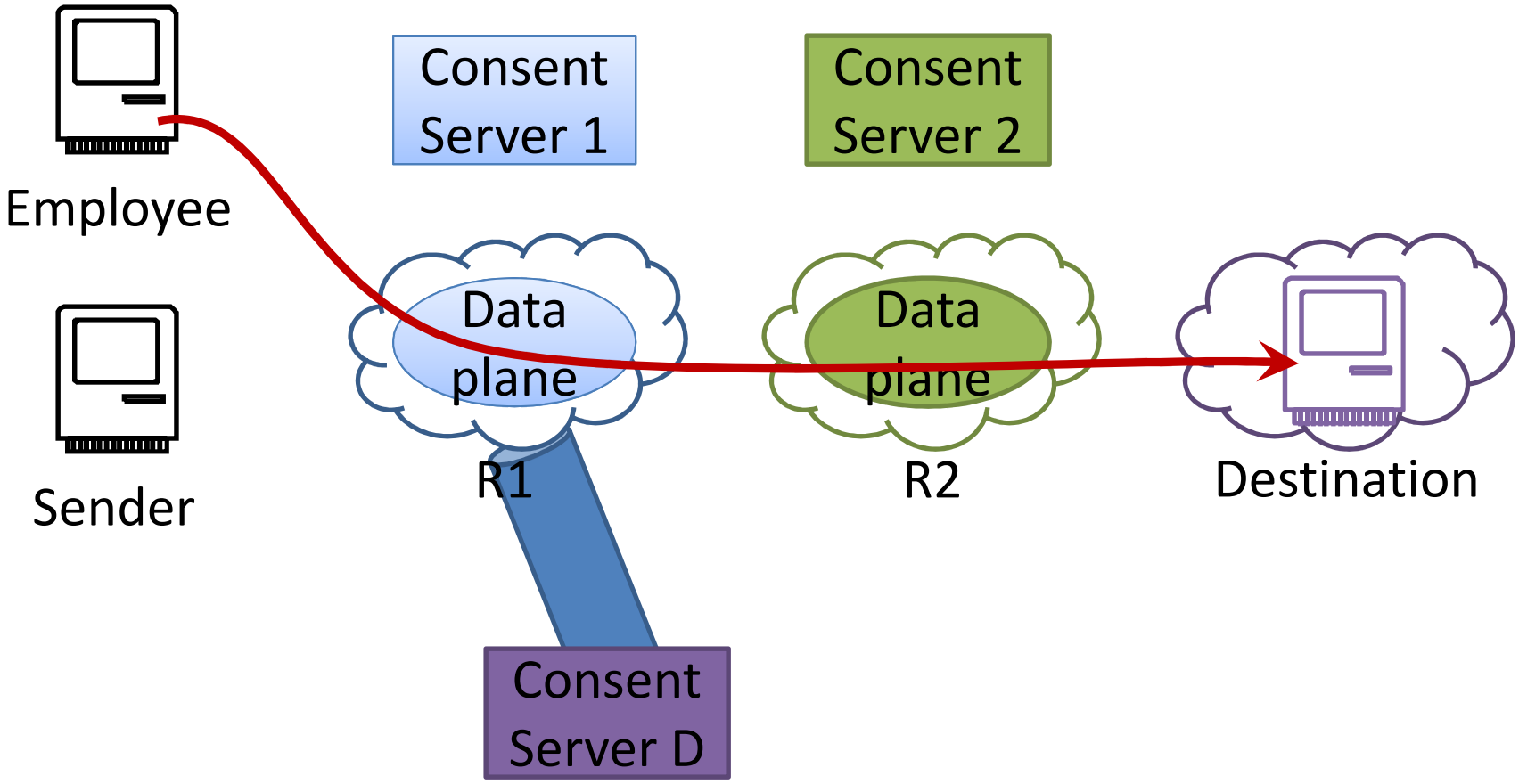
Example use: preventing denial-of-service



Consent server can be moved to where bandwidth is plentiful

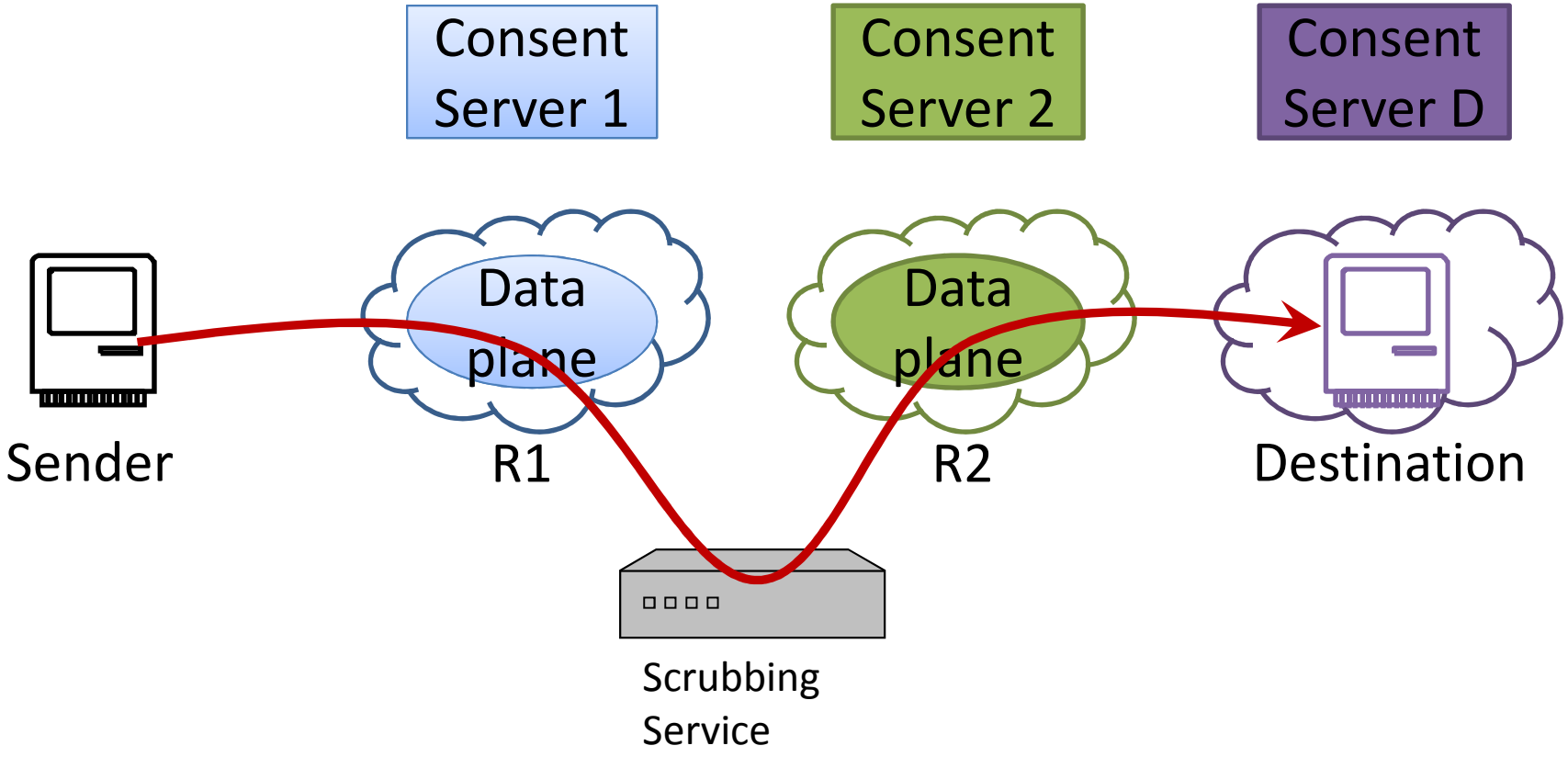
e.g. DoS prevention specialist

Example use: preventing denial-of-service



Employees can be given special keys to mint their own PoCs and not have to access a consent server

Example use: Off-site scrubbing service



- Consent is only granted if path goes through middlebox
- First honest realm drops the packet if middlebox not actually passed

Other uses

- Multipath
- QoS
- Billing support
- Access delegation
- …

Beyond this talk

- More data plane issues:
  - Bootstrapping (consent to get consent)
  - Key management/expiry/compromises
  - Network failures
  - Crypto details
- Pluggable control plane
  - Finding legal paths (routing)
  - Control delegation details
- Other issues:
  - Incremental deployment/benefit

Further Work

- More general and powerful policy engines
- Replay attacks
- Corner case attacks:
  - Putting legal full path in packet but only using prefix of the path.
- Route dissemination and other control plane overheads
- New business and economic models

Summary

- Policy framework for future Internet
- Principle of consent: Give all entities along a path control over path.
- ICING enables pluggable policy engines
- ICING is flexible, evolvable, and general