HIP Research Group Activities and Roadmap

Pekka Nikander, Ericsson Research Nomadic Lab
Tom Henderson, Boeing Phantom Works

Presentation outline

- HIP in a nutshell
  - Positioning and WG work
- A potential HIP roadmap
  - RG deployment plan
- Current activities
  - RG work
- Concluding remarks

Presentation outline

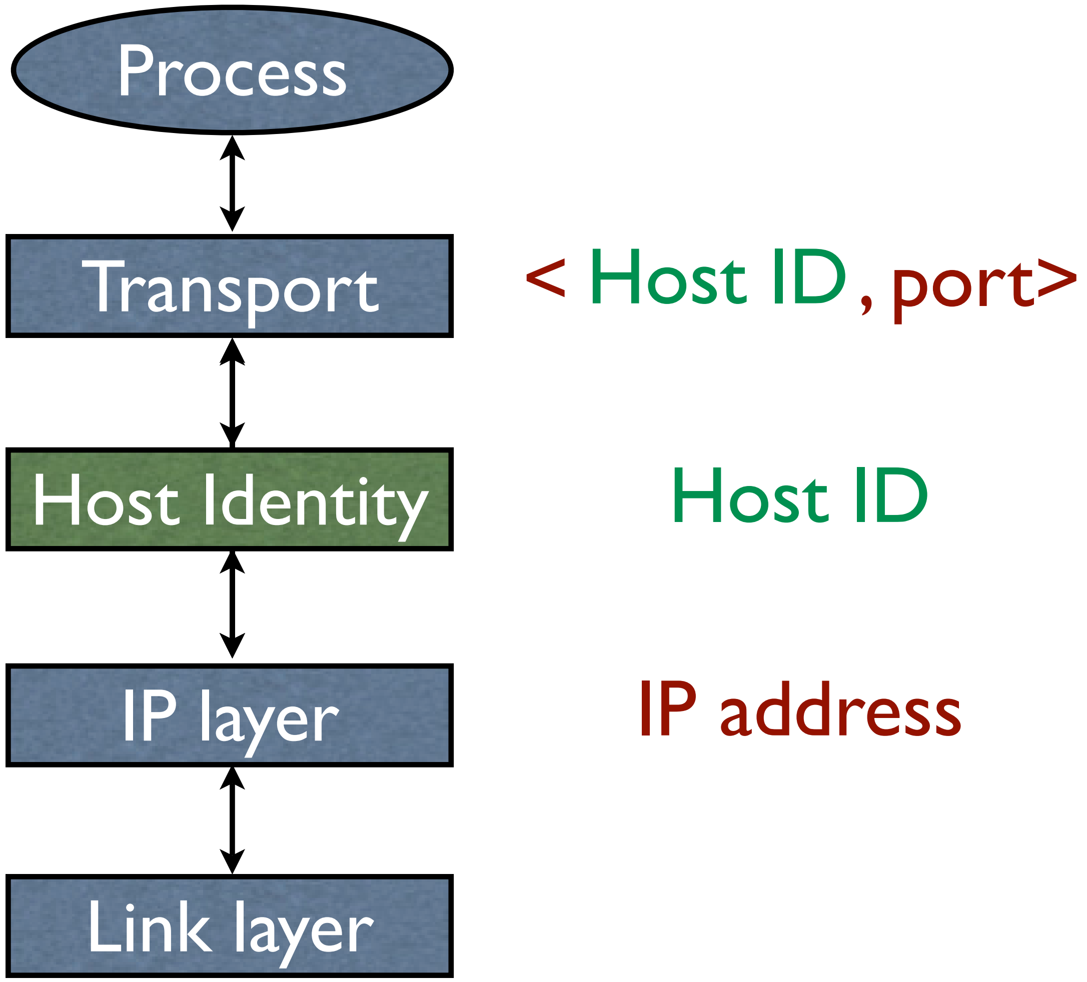
- HIP in a nutshell
  - What is HIP?
  - A brief history of HIP
  - Motivation; related WGs and RGs
  - WG work summary
- A potential HIP roadmap
- Current activities
- Concluding remarks

What is HIP?

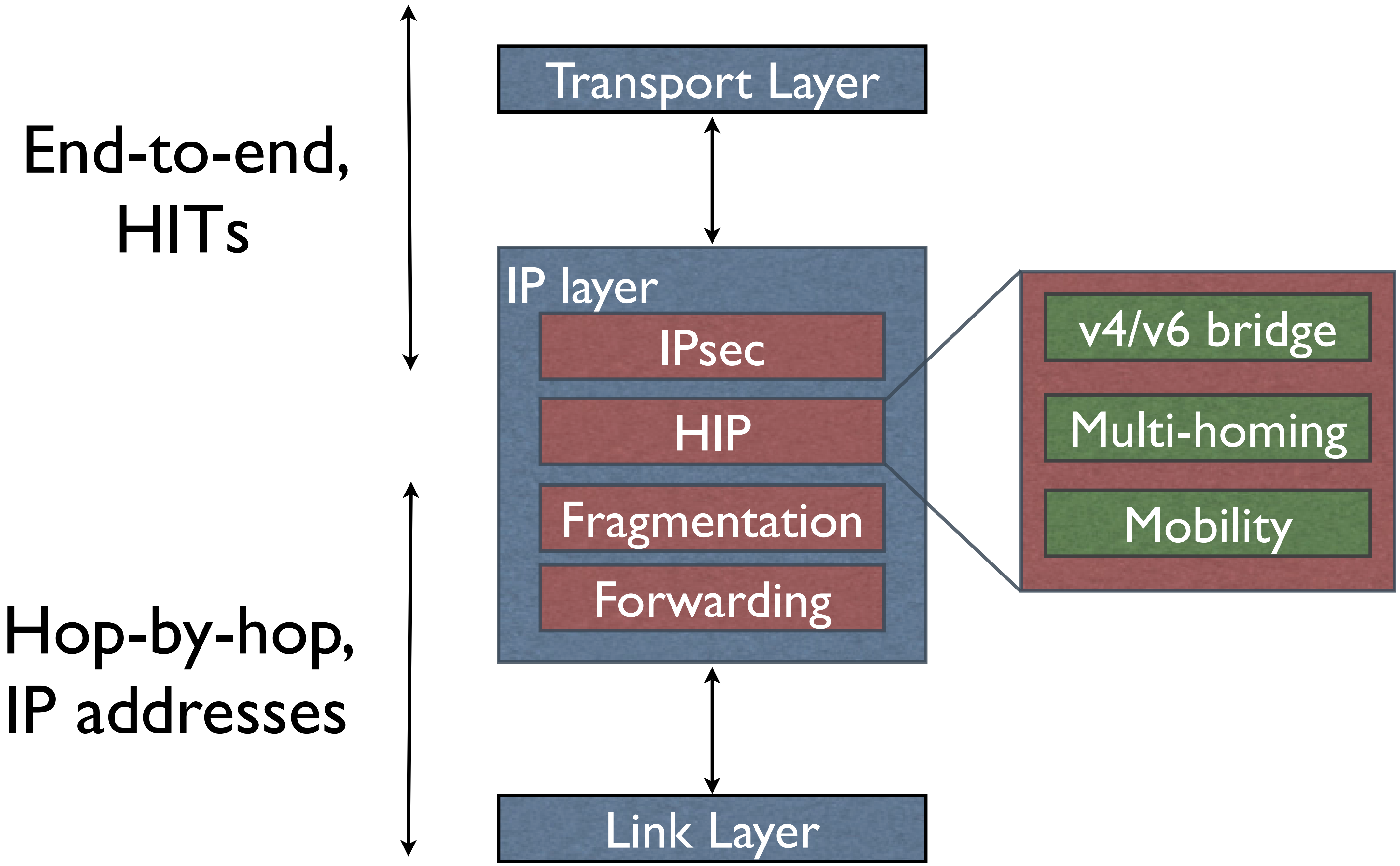
- HIP = Host Identity Protocol
- A proposal to separate identifier from locator at the network layer of the TCP/IP stack
  - A new name space of public keys
  - A protocol for discovering and authenticating bindings between public keys and IP addresses
    - Secured using signatures and keyed hashes

The Idea

- A new Name Space of Host Identifiers (HI)
  - Public crypto keys!
  - Presented as 128-bit long hash values, Host ID Tags (HIT)
- Sockets bound to HIs, not to IP addresses
- HIs translated to IP addresses in the kernel

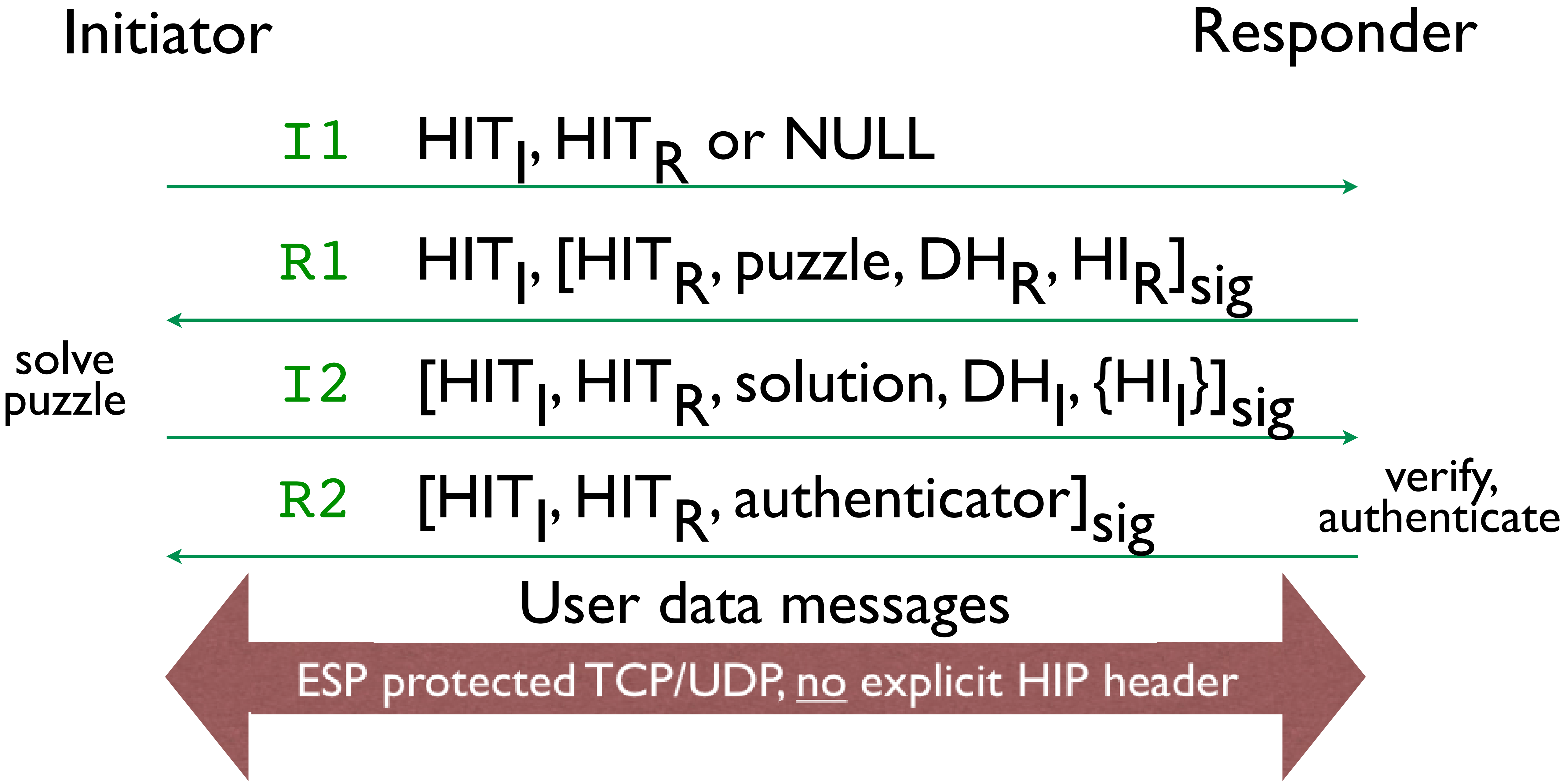


More detailed layering



Base exchange

- Based on SIGMA family of key exchange protocols



Initiator

Responder

I1 HIT_I, HIT_R or NULL

R1 HIT_I, $[HIT_R$, puzzle, DH_R, $HI_R]_{sig}$

solve puzzle

I2 $[HIT_I$, HIT_R, solution, DH_I, $\{HI_I\}]_{sig}$

R2 $[HIT_I$, HIT_R, authenticator$]_{sig}$

verify, authenticate

User data messages

ESP protected TCP/UDP, no explicit HIP header

draft-ietf-hip-base-02.txt, draft-jokela-hip-esp-00.txt

Other core components

- Per-packet identity context
    - Indirectly, through SPI if ESP (or SRTP) is used
    - Directly, e.g., through an explicit shim header
- A mechanism for resolving identities to addresses
    - DNS-based, if FQDNs used by applications
    - Or distributed hash tables (DHTs) based

A Brief History of HIP

- 1999 : idea discussed briefly at the IETF
- 2001: two BoFs, no WG created at that time
- 02-03: development in the corridors
- 2004: WG and RG created
- Now: base protocol more or less ready
  - Four interoperating implementations
- More work needed on mobility, multi-homing, NAT traversal, infrastructure, and other issues

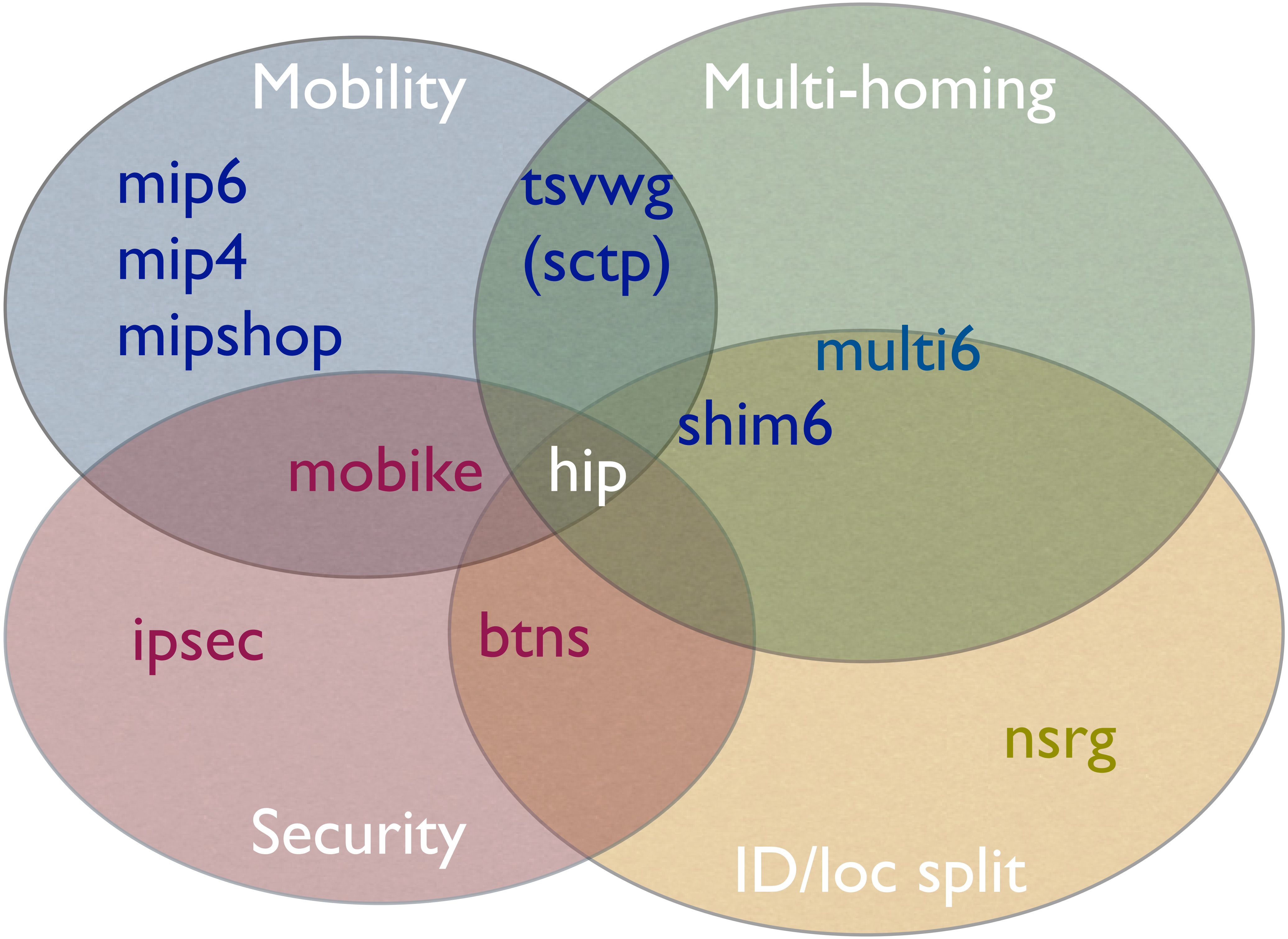
Motivation

- Not to standardise a solution to a problem
  - No explicit problem statement
- Exploring the consequences of the id / loc split
  - Try it out in real life, in the live Internet
- A different look at many problems
  - Mobility, multi-homing, end-to-end security, signalling, control/data plane separation, rendezvous, NAT traversal, firewall security, ...

Motivating architectural factors

- A “reachability” solution across NATs
    - New “waist” for the protocol stack
- Built-in security
    - Implicit channel bindings
        - `connect(HIT)` provides a secured connection to the identified host
    - Puzzle-based DoS protection
- Integrated mobility and end-host multi-homing

Related WGs and RGs



WG summary

- HIP WG is chartered to produce experimental RFCs:
    - Base protocol, use of ESP
    - Mobility and multi-homing
    - DNS resource record(s)
    - Registration protocol, (simple) rendezvous server
- However, we need to understand the implications of deploying HIP on a large scale
    - Changes to hosts and host management
    - Additional network infrastructure
- This latter topic is the focus of the HIP RG

Presentation outline

- HIP in a nutshell
- A potential HIP roadmap
  - Initial exploration
  - Early infrastructure
  - Enhanced Infrastructure
  - Early markets: HIP as a vertical solution
- Current activities
- Concluding remarks

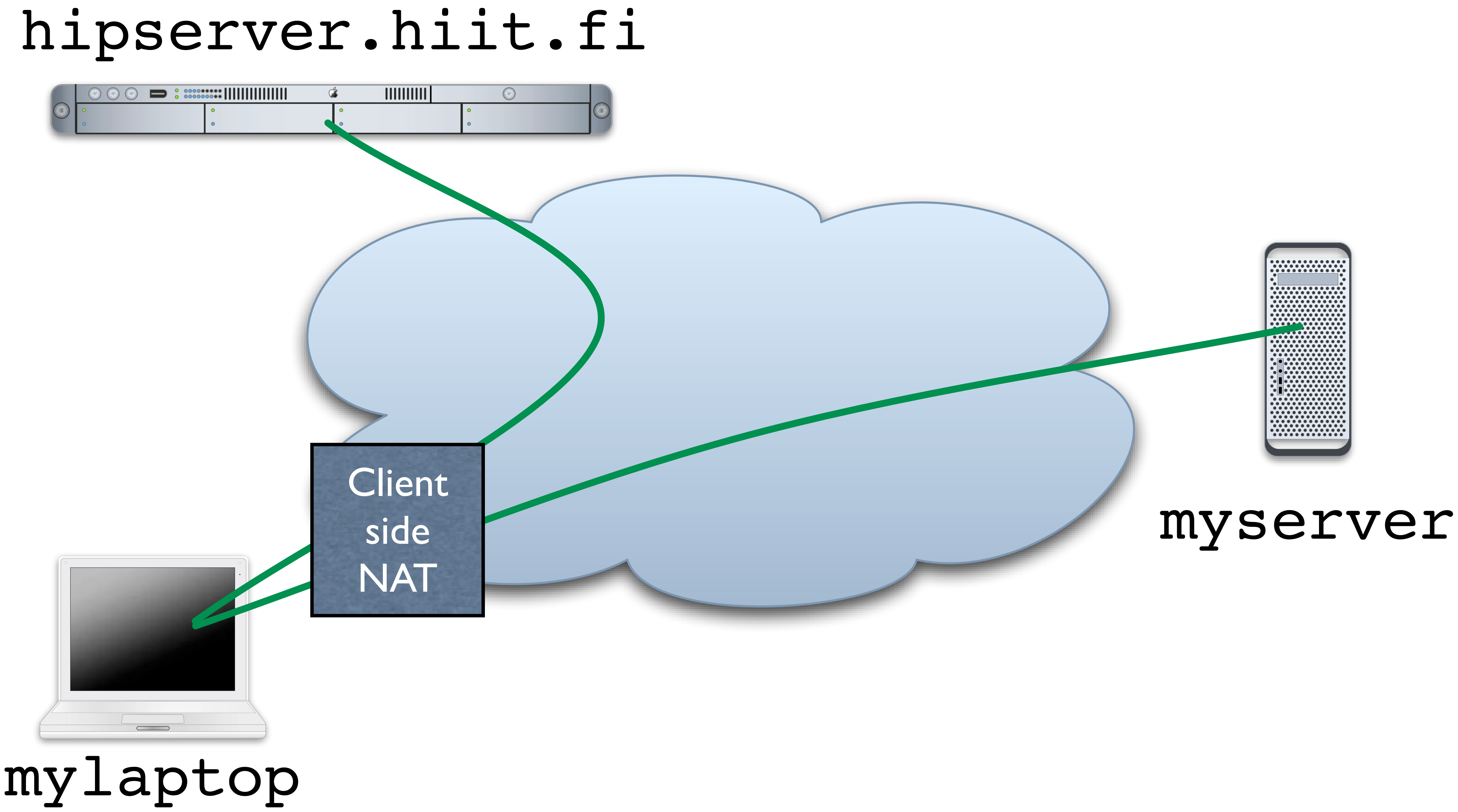
Initial exploration

- Pair-wise host-to-host deployment
  - e.g. my laptop and my personal server
- HITs typically stored in `/etc/hosts`

```
192.0.2.1 myserver
43bc:4521:4933:956c:3445:956d:ed23:3420 myserver
```

- Initial public test servers in the Internet
  - `hipserver.hiit.fi`

Initial exploration



Initial exploration: Requirements

- Host:
  - Install HIP on the host operating system
    - Linux: HIPL or Boeing HIP
    - BSD: HIP4BSD (FreeBSD; MacOS X soon)
    - Windows: Boeing HIP (cygwin based)
  - Configure HITs in `/etc/hosts`
  - Configure applications to refer to HITs
- Network: none

Initial exploration: Benefits

- End-to-end security between client and server
  - Trust based on static configuration
- Client mobility and multi-homing
  - Even across IPv4 / IPv6 boundaries
- IPv4 / IPv6 API-level interoperability
- Protection against CPU / memory DoS attacks
- Soon: Client-side NAT traversal
  - For plain client–server TCP / UDP protocols

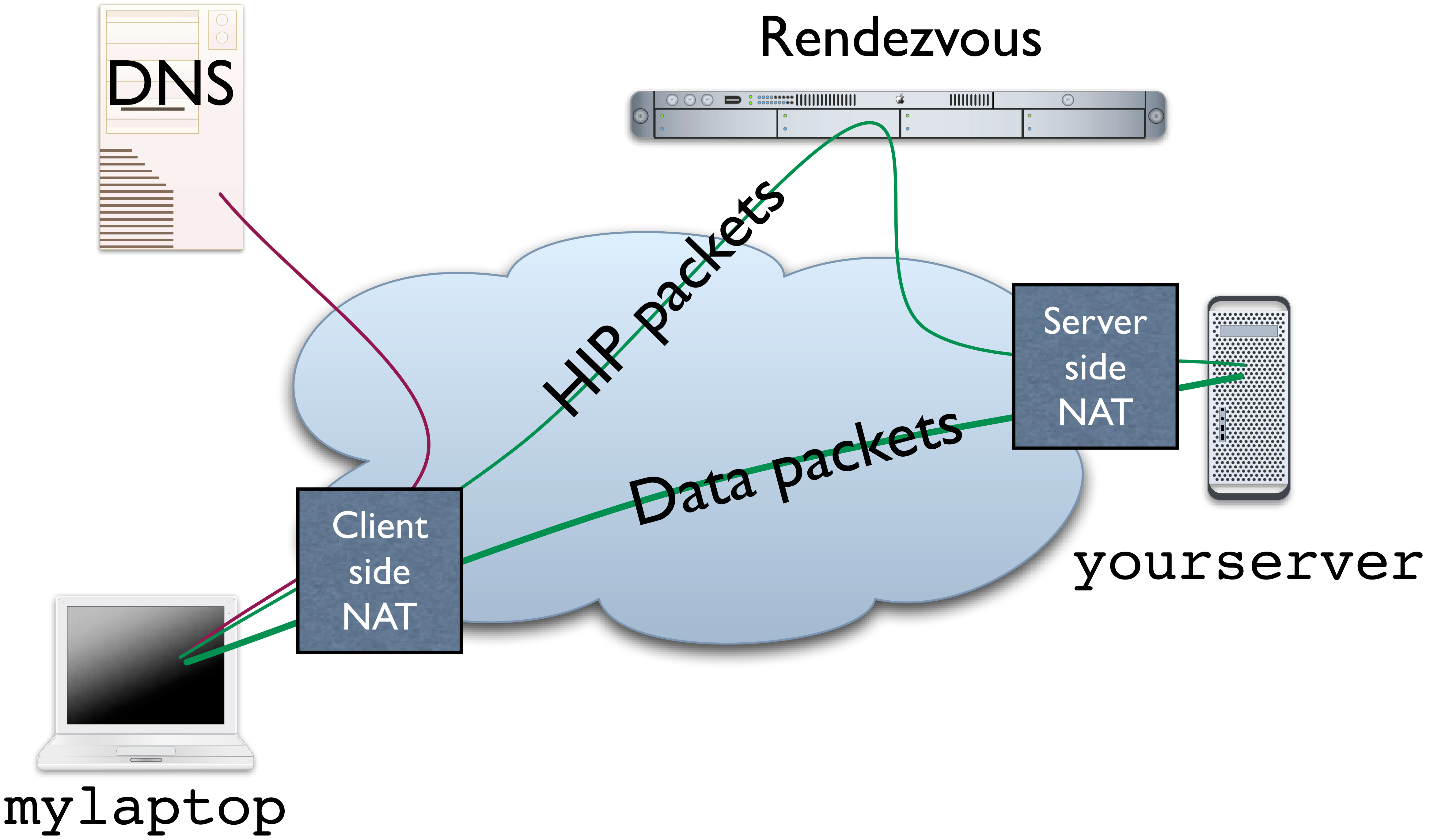
Initial exploration: Challenges

- Per-host management of a new name space
  - Policy configuration
    - Semantics for unsuccessful handshakes
  - Management of keys and address bindings
  - Privacy management
- Address resolution from HIT to IP address without any infrastructure
  - Must be explicitly configured

Early infrastructure

- Pair-wise deployment between early adopters
  - e.g. my laptop and your experimental server
- Store HITs in the DNS as AAAA RRs
  - Look like non-routable IPv6 addresses
  - Returned as the last entry in an RR set
- Experimental rendezvous (Hi^3) at PlanetLab
  - Infrastructure for passing HIP packets

Early infrastructure



Early infrastructure: Requirements

- Host:
  - No new significant requirements
  - Maybe an update of the HIP software
- Infrastructure on the network:
  - Store HITs to DNS as AAAA records
  - Install experimental rendezvous servers
- Routers and NATs:
  - no changes

Early infrastructure: (Additional) benefits

- *Opportunistic* security between participants
  - Perhaps build trust with DNSSEC
- Simultaneous mobility; i.e., mobile servers
- Increases the cost of some flooding DoS attacks
  - Potential attacker needs to solve the HIP puzzle before getting the real IP address
- NAT traversal for both client and server
  - Unlikely to work for symmetric NATs

Enhanced infrastructure

- Internet-wide experimental deployment
  - Stable rendezvous service
  - Store HITs in the DNS using new RRs
- Benefits as before but larger audience
- Results to be reported in HIP RG experiment report
  - Input to the IETF community

Markets take over: HIP on selected vertical markets

- Potential markets
  - Multi-homed road warriors
  - Operations and management
  - Military or dual-use systems
  - High-availability systems
  - Mobile public networks
    - e.g., municipal 802.11 networks

Presentation outline

- HIP in a nutshell
- A potential HIP roadmap
- Current activities
  - NAT traversal or layer 3.5 connectivity
  - Upper layer identifiers
  - H*i*^{3} and other DHT-based rendezvous
  - Separating control and data traffic
- Concluding remarks

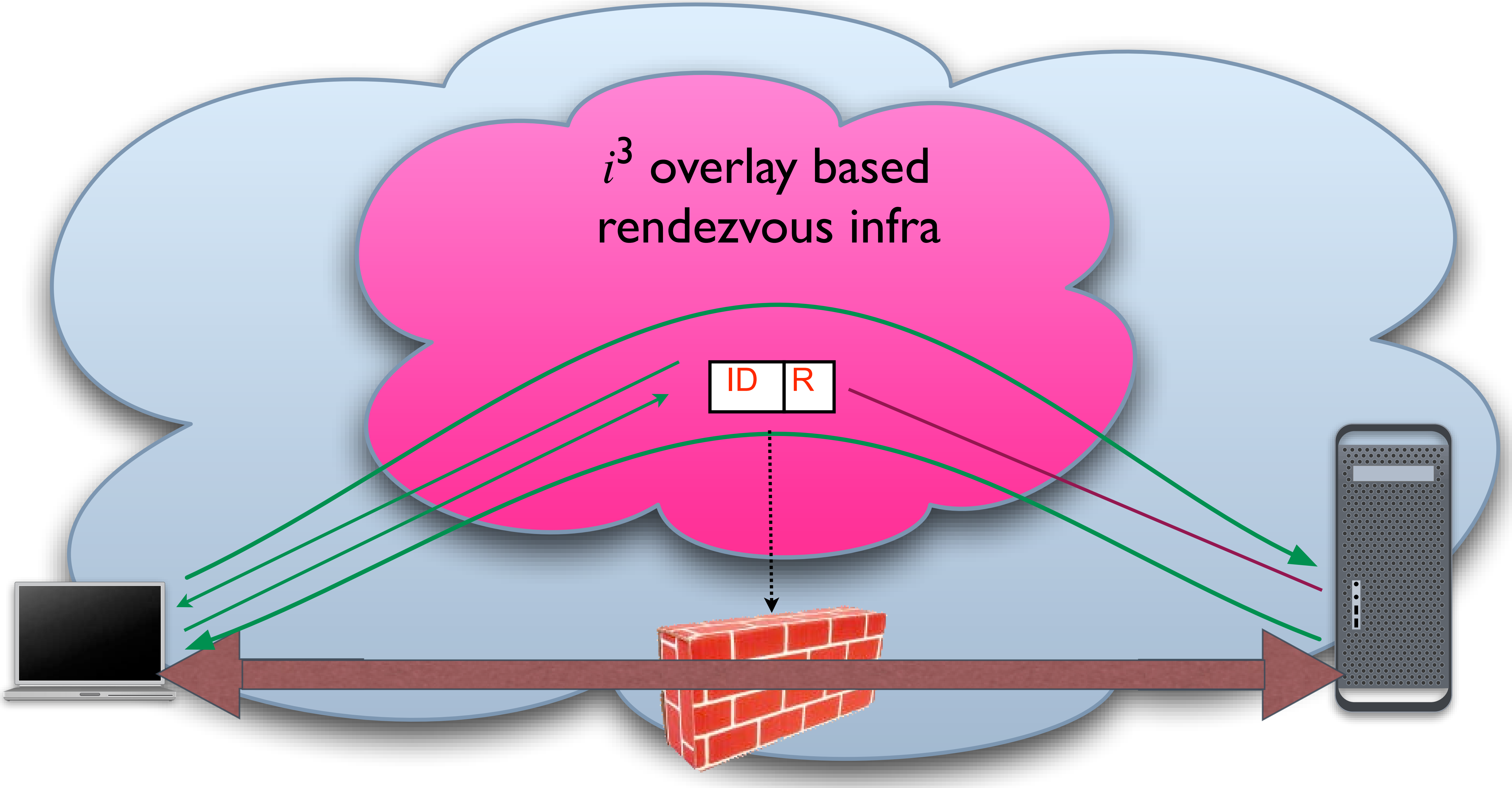
NAT traversal

- Legacy NAT traversal
  - Apply ideas from STUN/ICE/STUNT... to HIP
  - UDP tunneling
  - Short term solution with a clear exit strategy
- SPI-NAT or architected NAT
  - Make NAT aware of HIP messages
  - Allow servers to register at the NAT
  - Learn mappings for HITs and ESP SPIs

Upper layer identifiers

- Backward compatible APIs
  - Current APIs form a major legacy asset
  - HIP allows almost all applications to continue unmodified (no recompilation required)
  - Q: Use HITs / IP addrs / both as the ULID?
- New APIs
  - Host vs. Session vs. Service identifiers?
  - Using delegation?

Hi^3 and DHT-based rendezvous



Separating control and data

- Originally HIP was tightly bound to ESP, using ESP as the data encapsulation protocol
- ESP split from the base specification
  - Allow other encapsulations in the future
  - Maybe even plain TCP / UDP w/ null encaps
- Fast / slow path separation at middle boxes
- Optionally different locators for control / data

Summary

- HIP WG producing components for experimental deployment:
  - base protocol, ESP, mobility & multi-homing, DNS, registration, rendezvous
- HIP RG preparing for real life experiments
- On-going RG work items:
  - NAT, ULIDs and APIs, H*i*^{3} / DHT based rendezvous, separation of control and data

Concluding remarks

- Base protocol ready for early exploration
  - Interoperating OSS implementations available
- Open questions looking for answers
  - Impact: on hosts, routers, other infra
  - Architectural questions: ULIDs, resolution, separation of control and data, ...
  - New functionality: DDoS protection, moving networks, MANETs, ...