



SmartSockets:

Solving the Connectivity Problems in Grid Computing



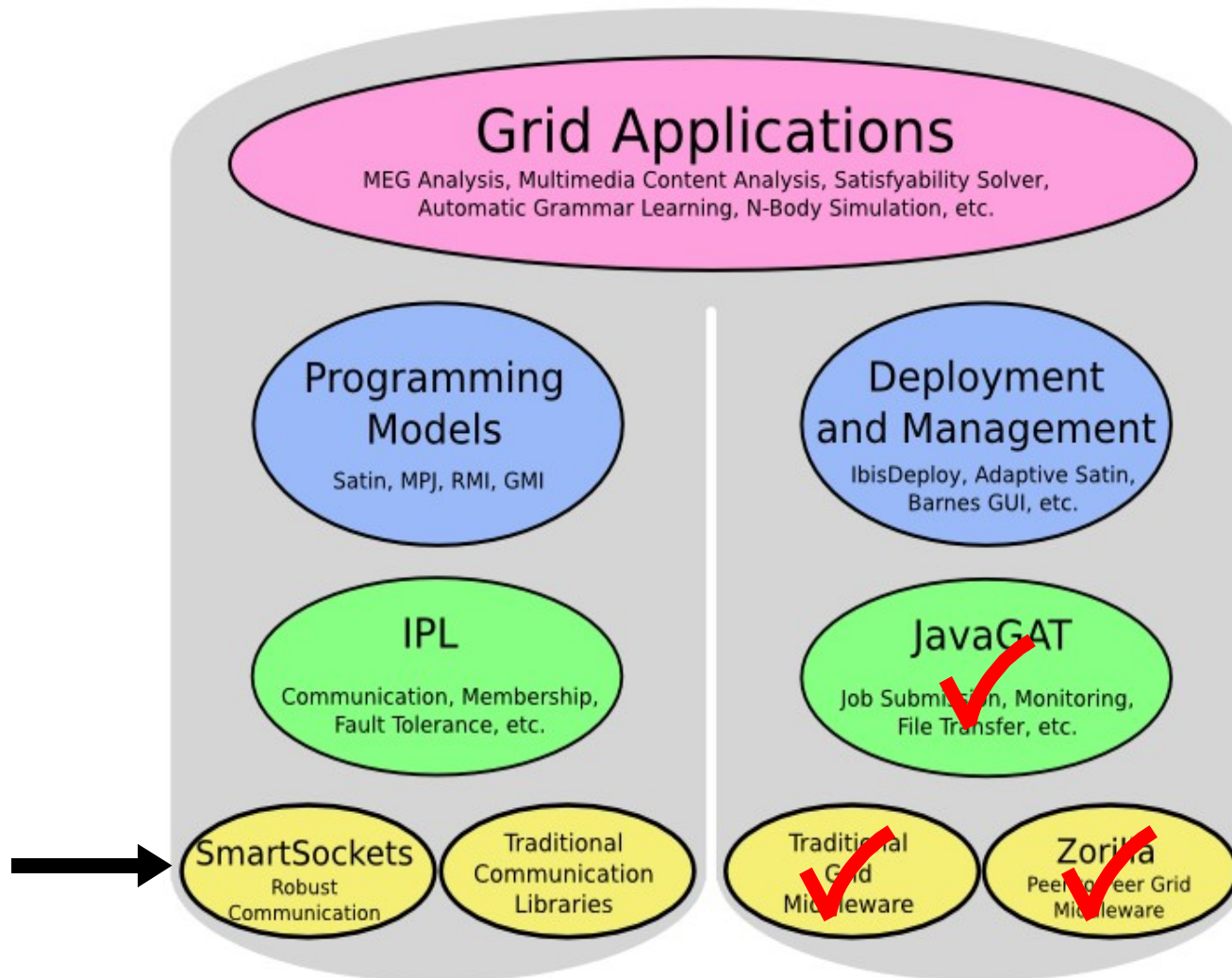
Jason Maassen
jason@cs.vu.nl

Until now...

- We looked at how Ibis makes deployment user friendly:
 - **JavaGAT** provides an easy-to-use API for the various flavours of Grid middleware
 - **Zorilla** provides a configuration free alternative to existing Grid middleware
- Next step:
 - **Communication**



Overview



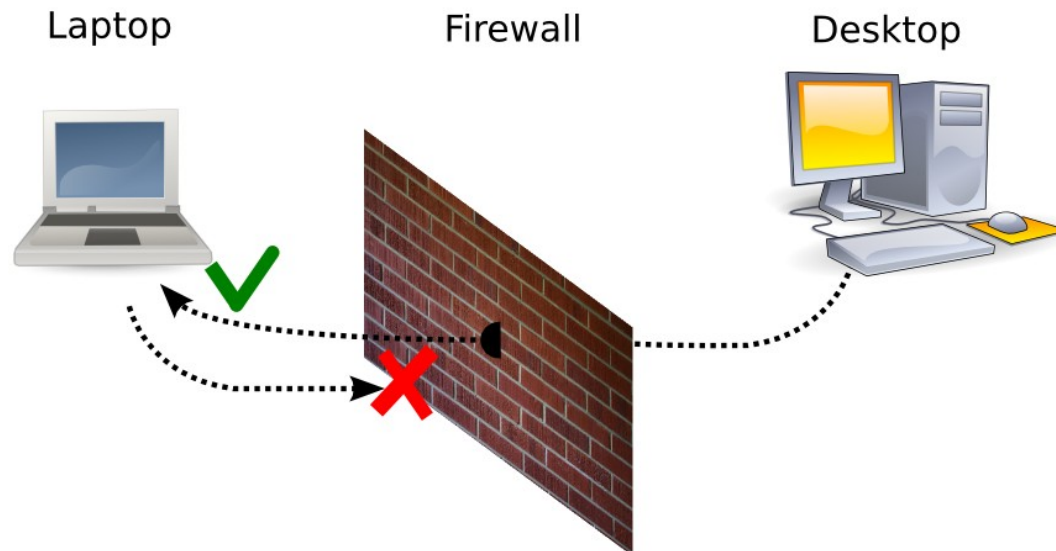
Communication is Difficult!

- Many sites have connectivity issues
 - Firewalls
 - Network Address Translation (NAT)
 - Non-routed networks
 - Multi homing
 - Mis-configured machines
 - ...
- This makes it hard to use a combination of sites



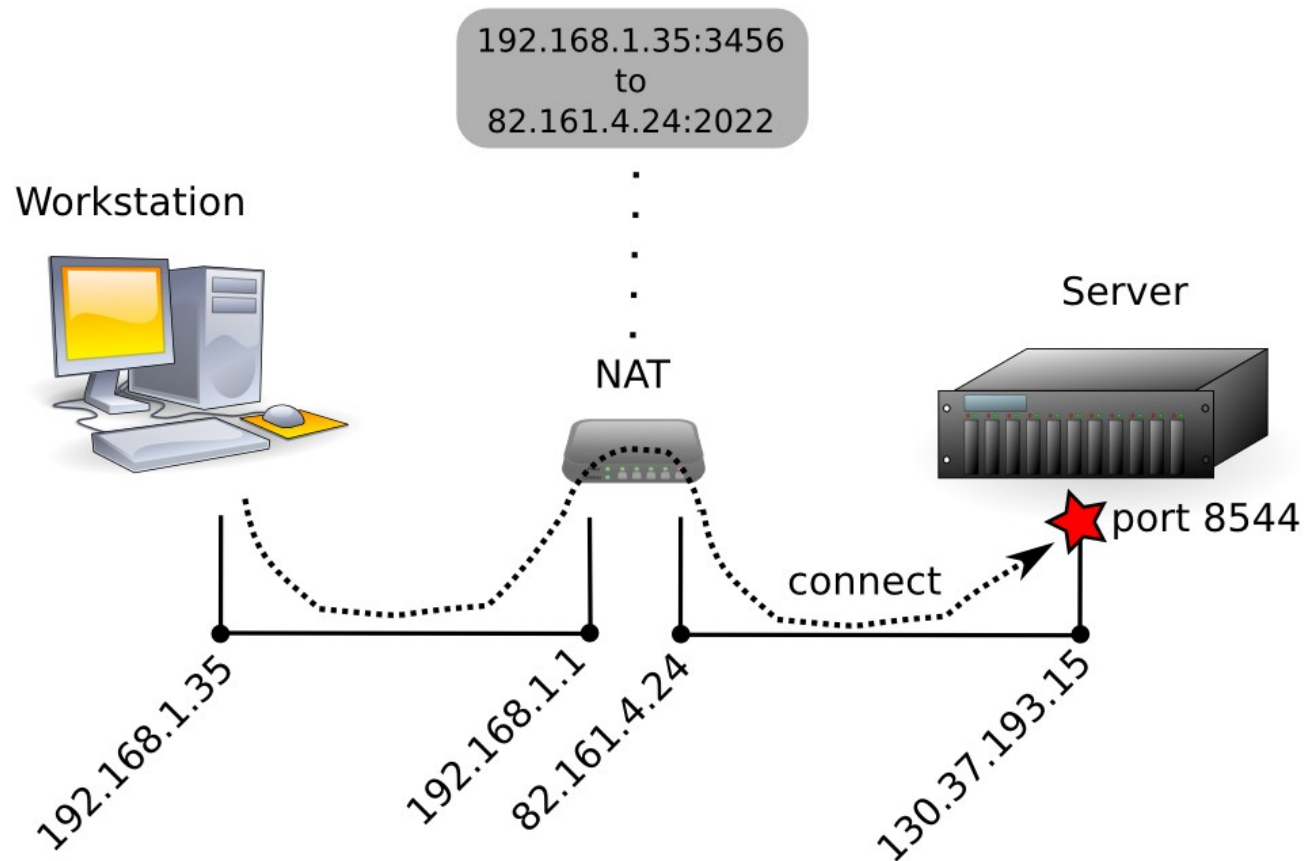
Problem 1: Firewalls

- Blocks 'inappropriate' traffic
 - Usually only blocks incoming traffic
 - Some also block outgoing traffic



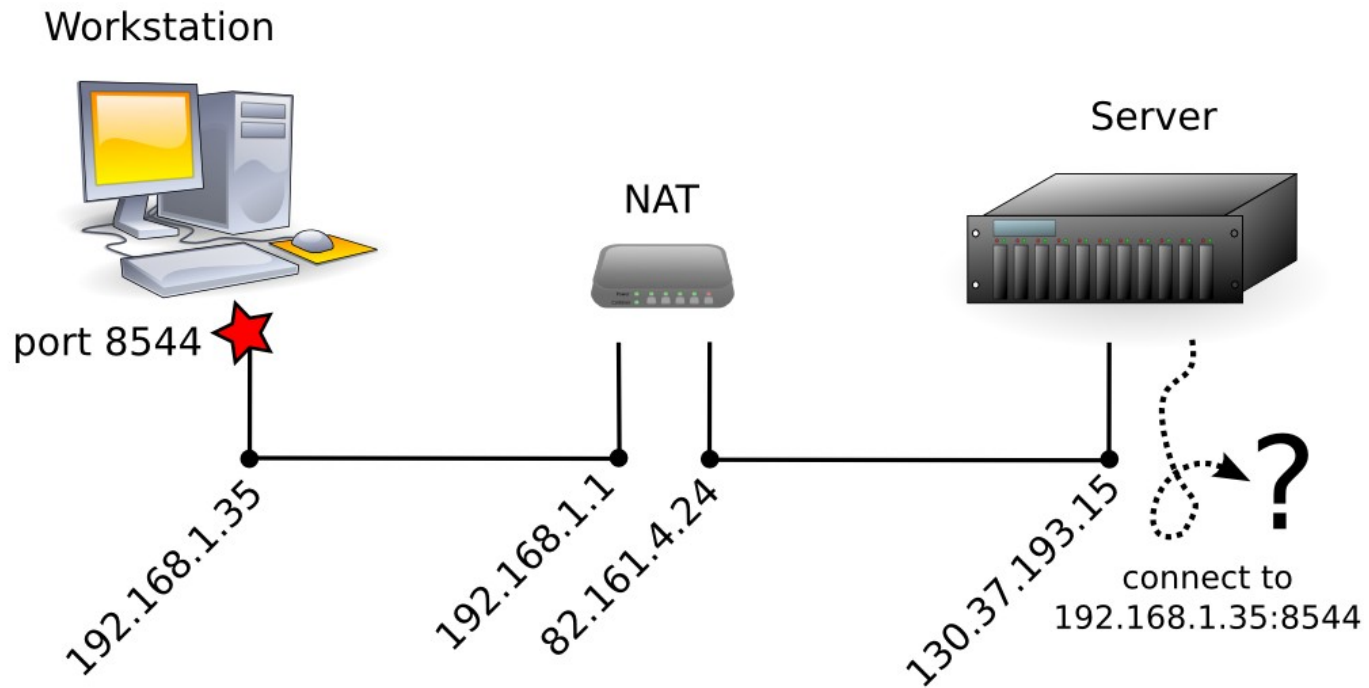
Problem 2: Network Address Translation

- Maps IP from one range into another
 - Works fine for outgoing connections



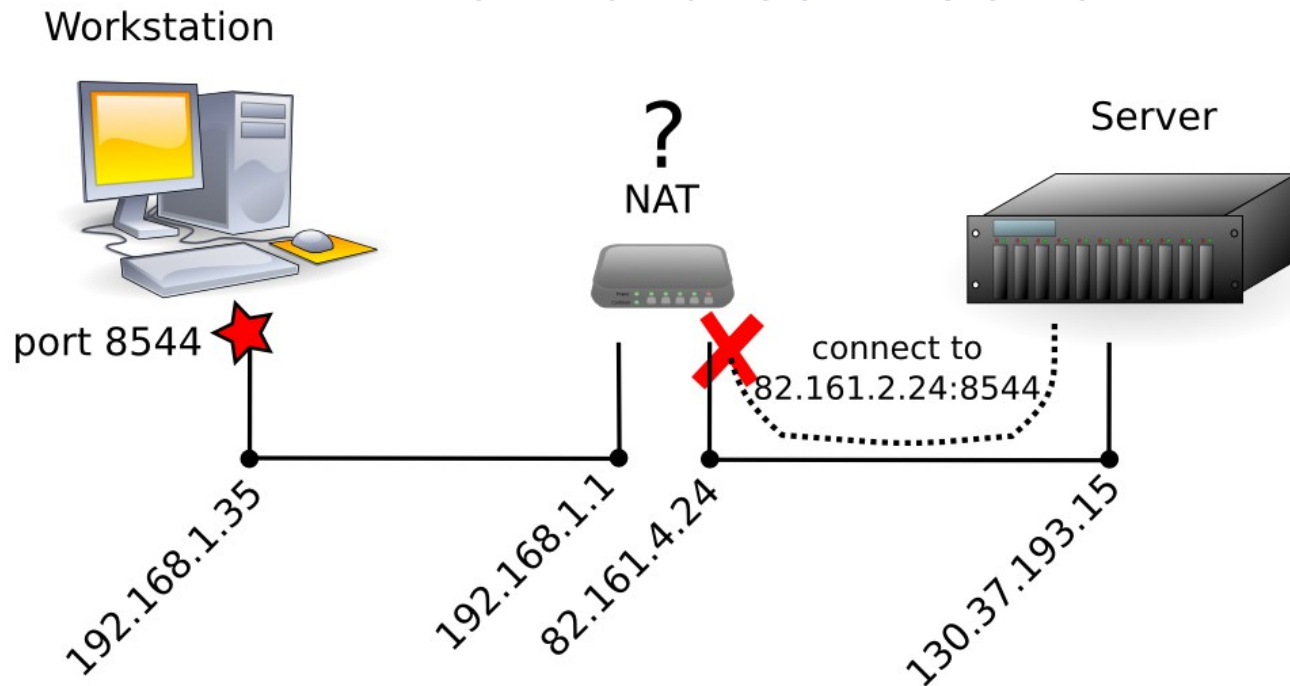
Problem 2: Network Address Translation

- Problem: incoming connections
 - Target address is invalid on internet



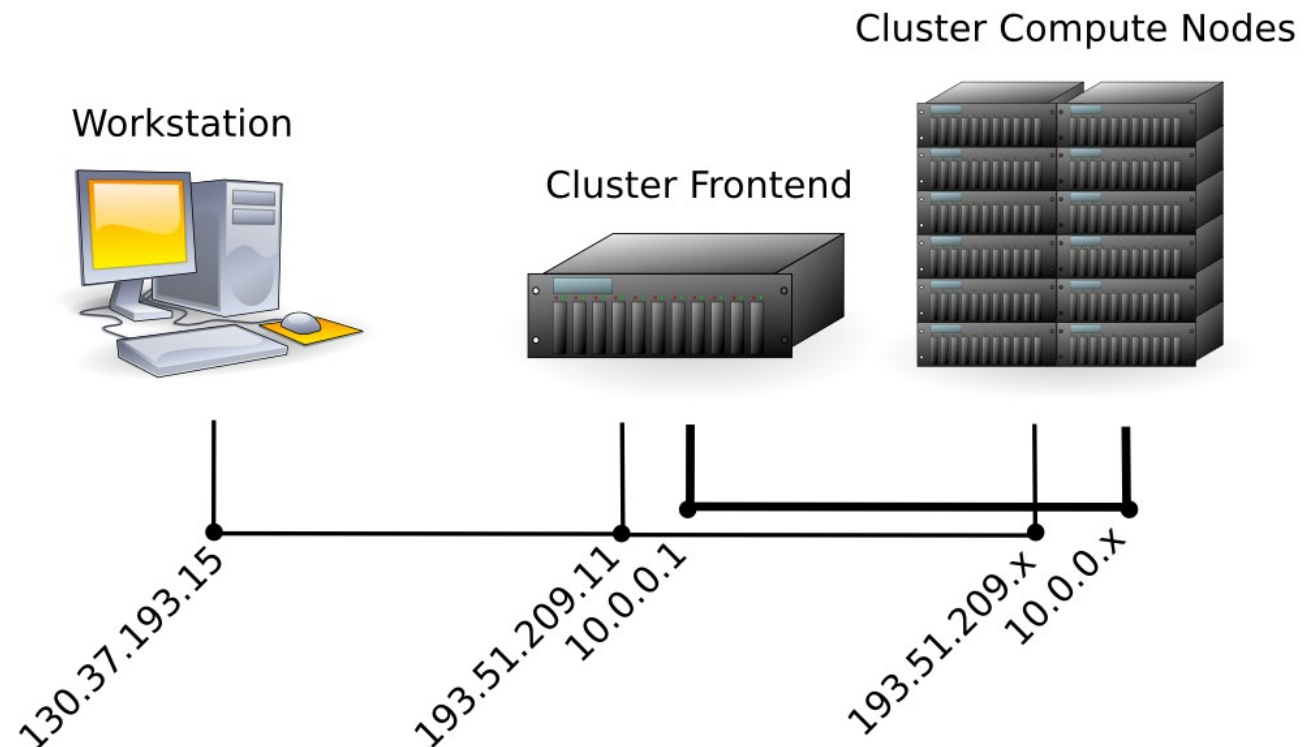
Problem 2: Network Address Translation

- Problem: incoming connections
 - Target address is invalid on internet
 - NAT device does not know where to forward connection



Problem 3: Multi Homing

- Some sites have multiple networks
 - The target address depends on the source of the connection

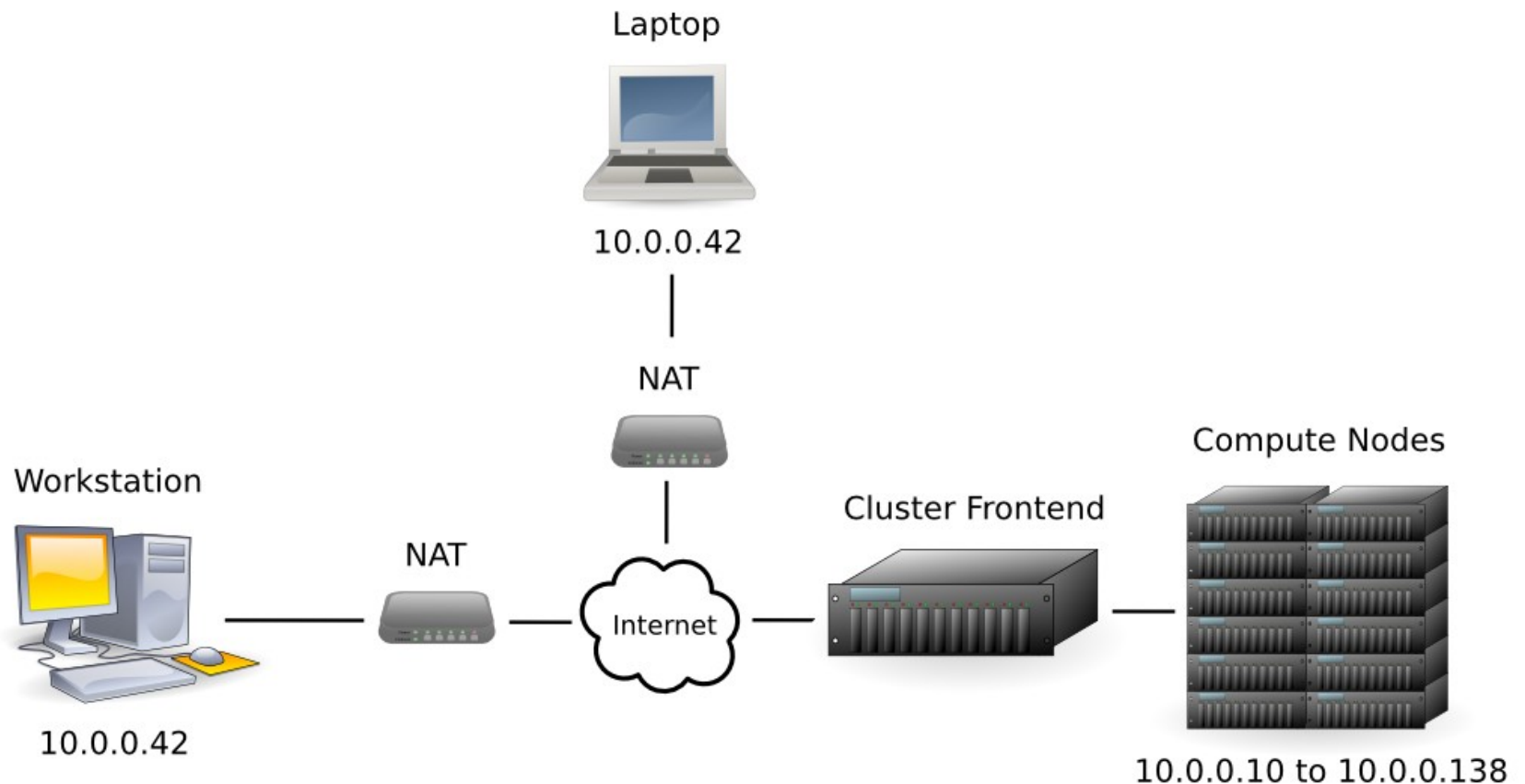


- Some sites do not route between the local network and internet
 - Only the frontend is reachable



Problem 5: Machine Identification

- NAT/non-routed networks lead to machine identification problems



Current Solutions

- There are many ad-hoc solutions:
 - Open port ranges in the firewall
 - Use port forwarding in NAT
 - Explicitly specifying IP addresses
 - SSH tunneling
 - ...
- These solutions work (sort of), but...



Current Solutions (Cont'd)

- It all up to the user!
 - What is wrong ?
 - How it can be solved ?
- A lot of work
 - Need help from system administration
 - May need changes to application
 - Needs to be re-done whenever the testbed changes!



SmartSockets

- The SmartSockets library
 - Offers a socket-like interface
 - Addressing is different
 - Detects connectivity problems
 - Tries to solve them automatically
 - With as little help from the user as possible
 - Integrates existing and several new solutions into one library
- User friendly connection setup!



Smart Addressing

- SmartSockets extends addressing
 - Not just a single IP:port combination
- Add all machine addresses
- Optionally add extra information
 - External address + port (for NAT)
 - UUID (if entire address is private)
 - SSH contact information



Addressing Examples

130.37.193.15-42611
public address

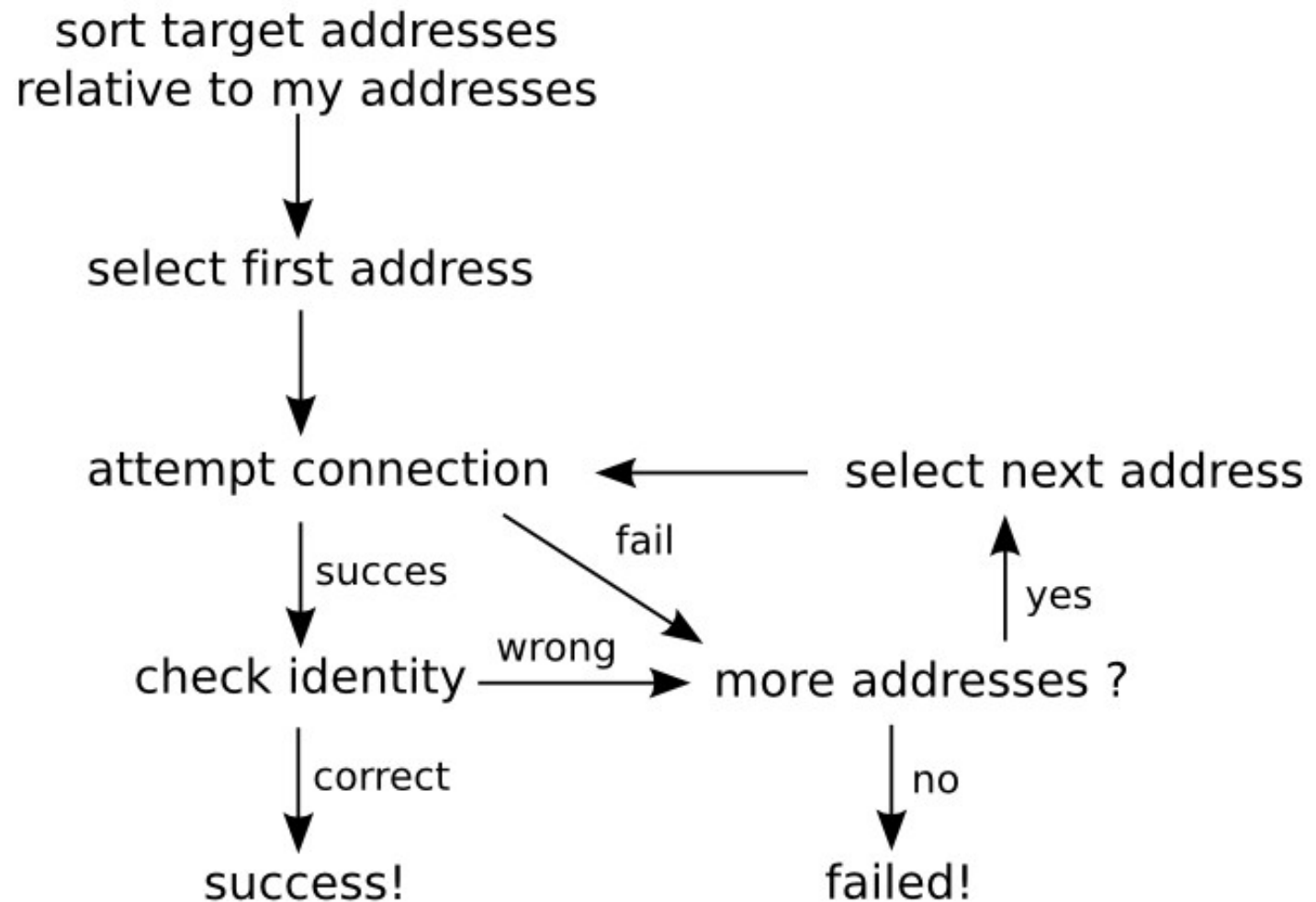
130.37.193.15-42611~jason
public address SSH user

130.37.197.201-42611/10.153.0.201/10.141.0.73-42611
public address private addresses

{82.161.4.24-20122}/192.168.1.36-42611#a6.e5.01.1a.ad.f1
external address with port forwarding private address UUID unique to host



Creating a Connection

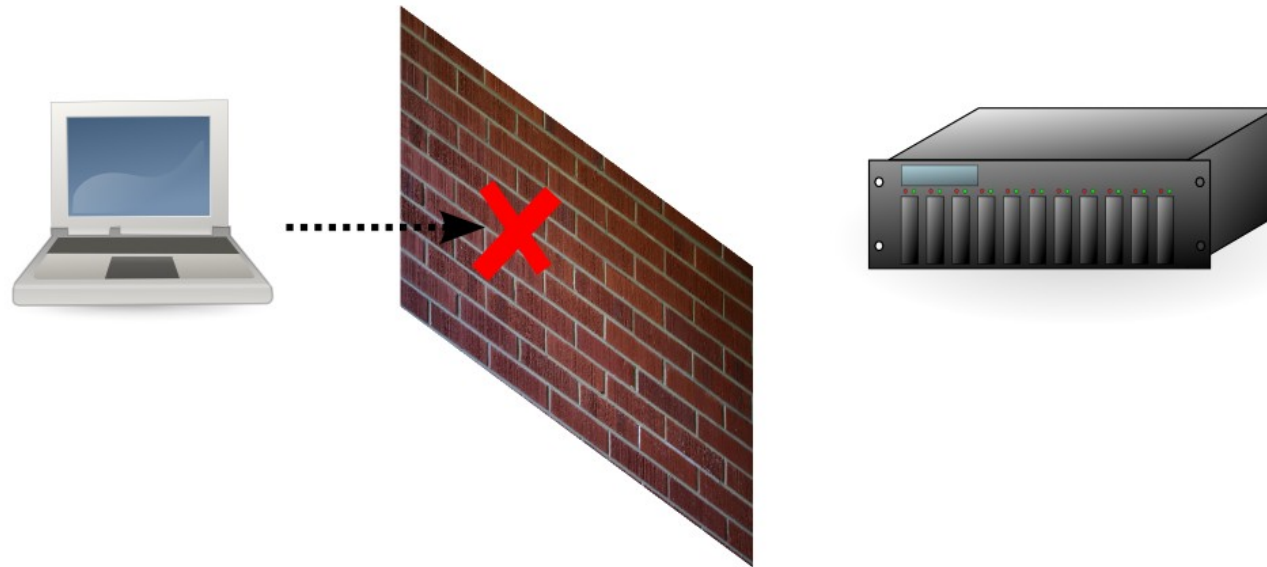


First Results

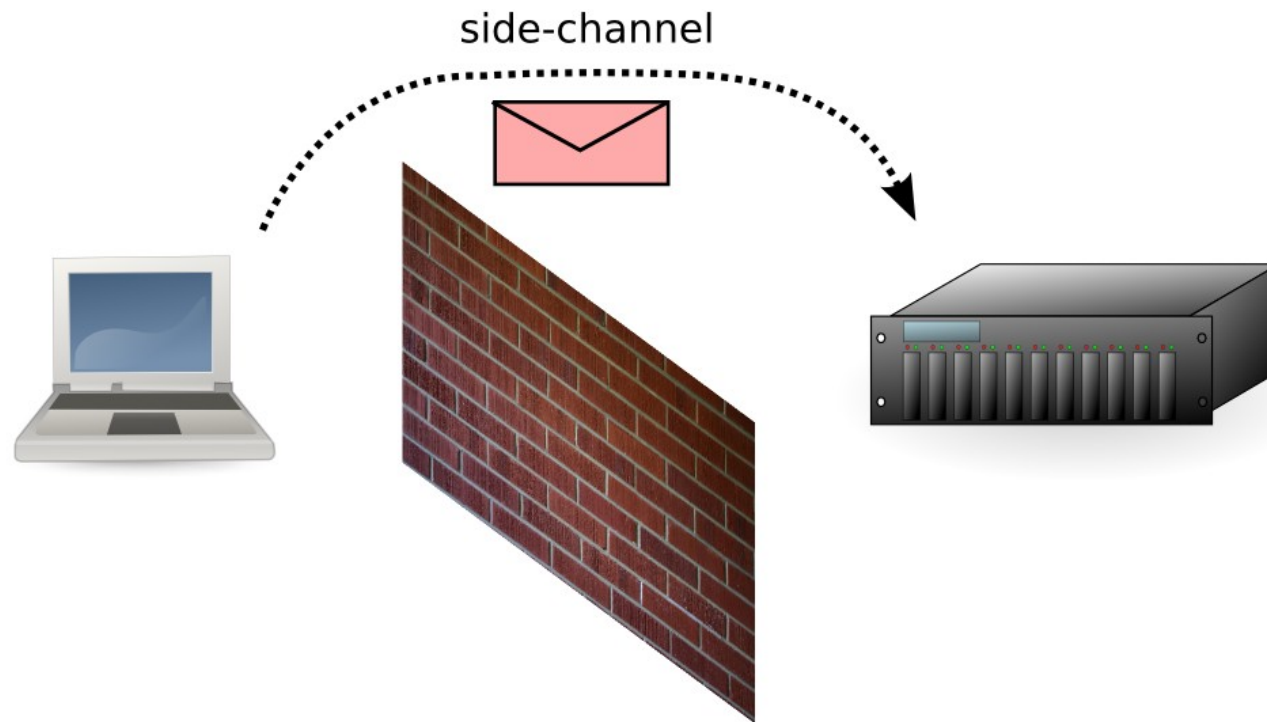
- This solves multi homing and machine identification problems (3 & 5)
- Assumes anyone can create a connection
 - This will not help when the target is behind a NAT / Firewall
- To solve this we need cooperation between machines....



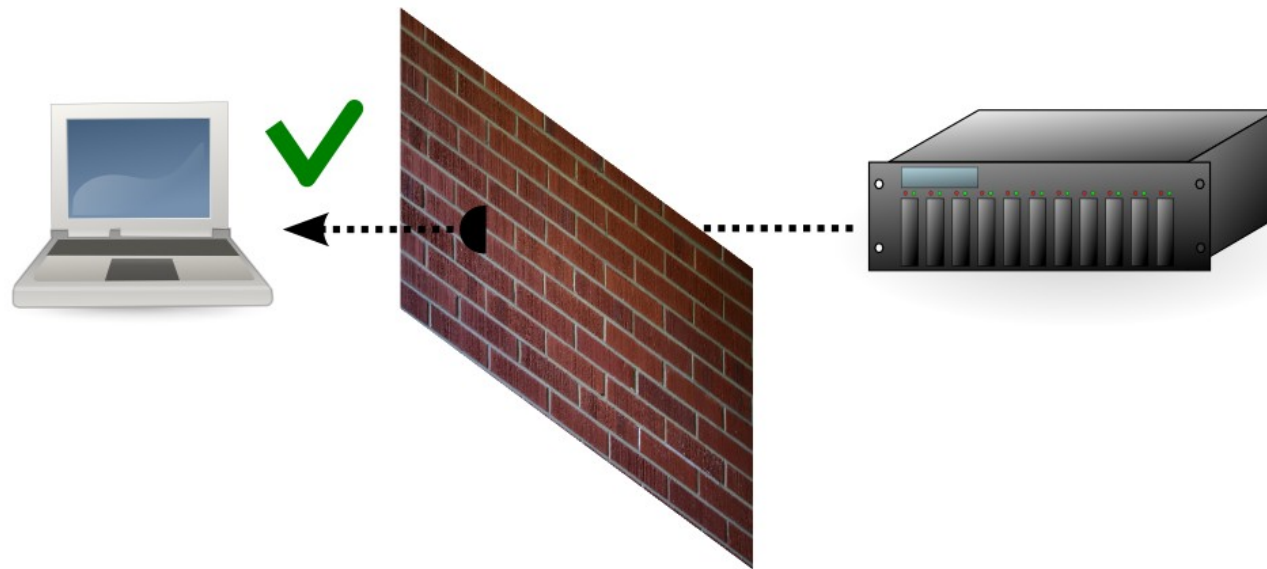
Cooperation



Cooperation



Cooperation



Side Channel

- SmartSockets uses network of *hubs* to implement a *side channel*
 - Support processes for the application
 - Started in advance
- Hubs are run on machines with 'more connectivity'
 - Such as cluster frontends, 'open' machines, etc.

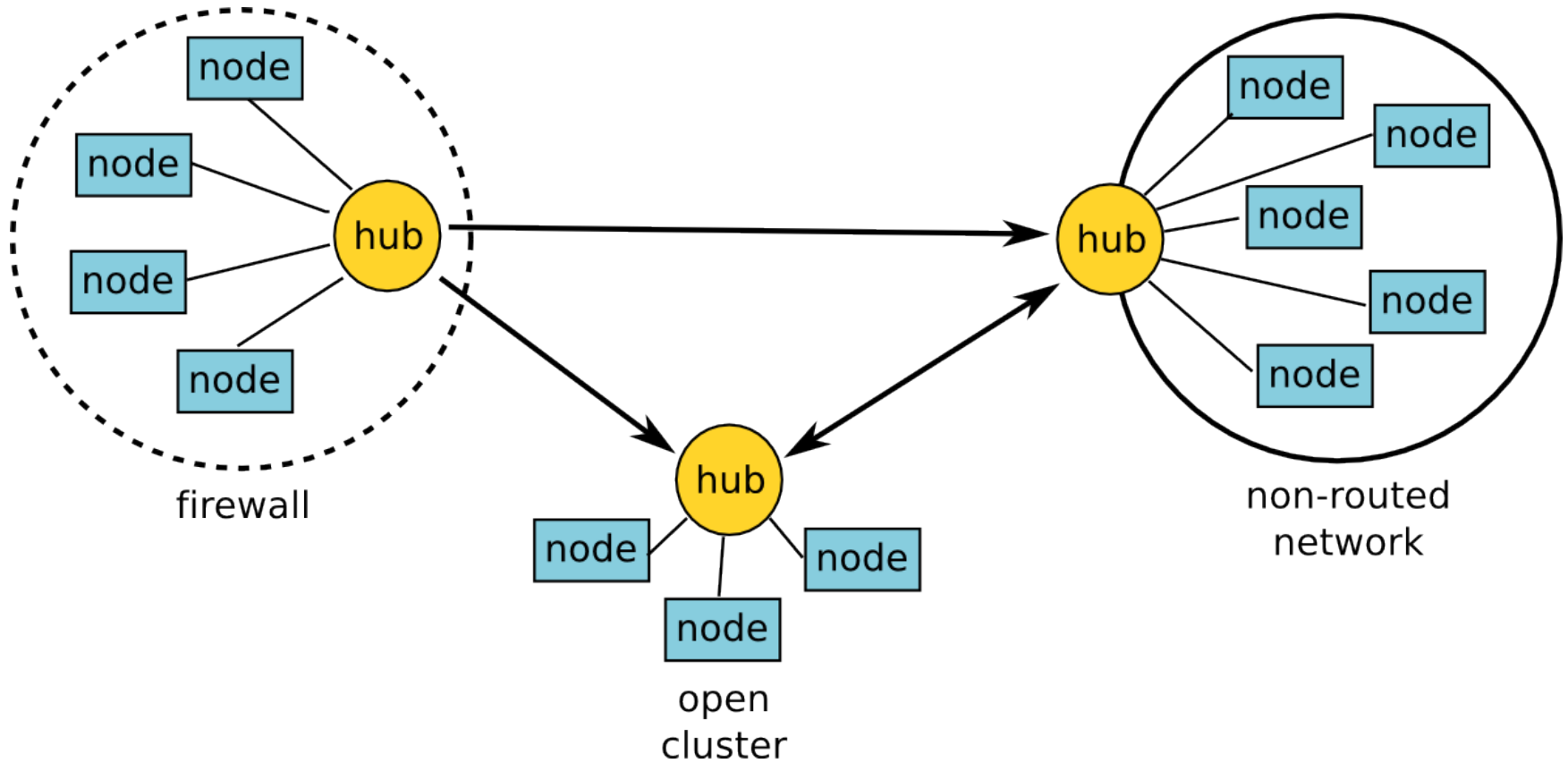


Hubs

- Hub connect to each other
 - Need to set up spanning tree (or better)
 - Use direct connections and SSH tunnels
 - Gossip information and client messages
 - Automatically discover new hubs and routes
- Clients connect to a 'local' hub
 - When needed, use network of hubs as side channel during connection setup



Example



Reverse

- Reverse direction of connection setup
 - Send message to target using hub
 - Wait for incoming (direct) connection
- Results in **direct connection**
 - Only connection setup is different
- Solves Firewall/NAT problems (1 & 2)
 - Assumes one side can create connection
 - Does not work with multiple firewalls or NATs, or non-routed networks



Splicing

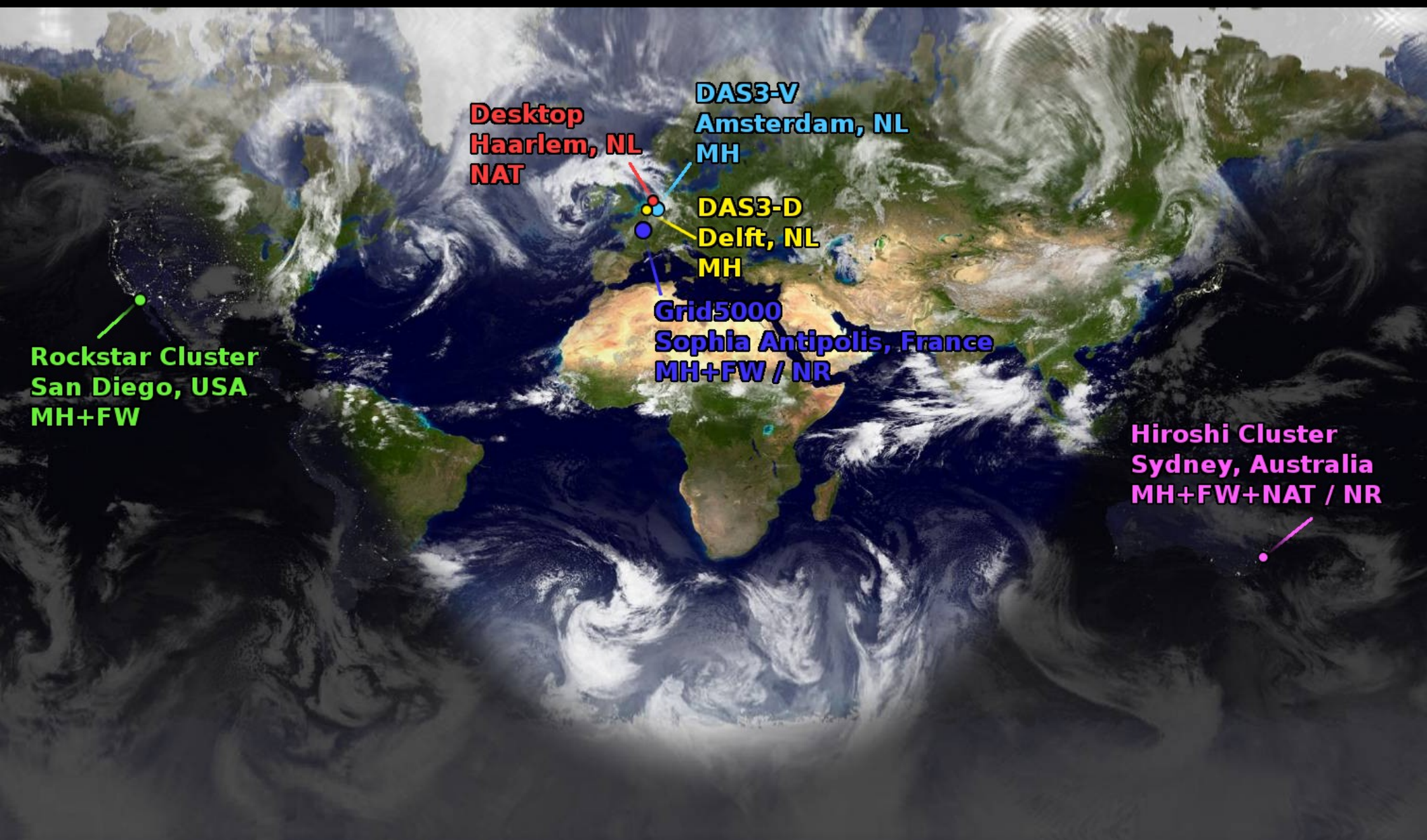
- Both sites simultaneously set up a connection
 - Connections meet in the middle
 - NAT requires port mapping prediction
- Results in direct connection
- Solves multi-firewall/NAT problems
 - Very sensitive to timing
 - Not guaranteed to work
 - Does not work for non-routed networks



Routed

- Create *virtual connection* using hubs
 - Forward all data over side channel
- Results in **indirect connection**
 - Usually lower performance
 - Still looks like a socket!
- Solves non-routed problem (4)
 - Also solves the multi-firewall/NAT case





Rockstar Cluster
San Diego, USA
MH+FW

Desktop
Haarlem, NL
NAT

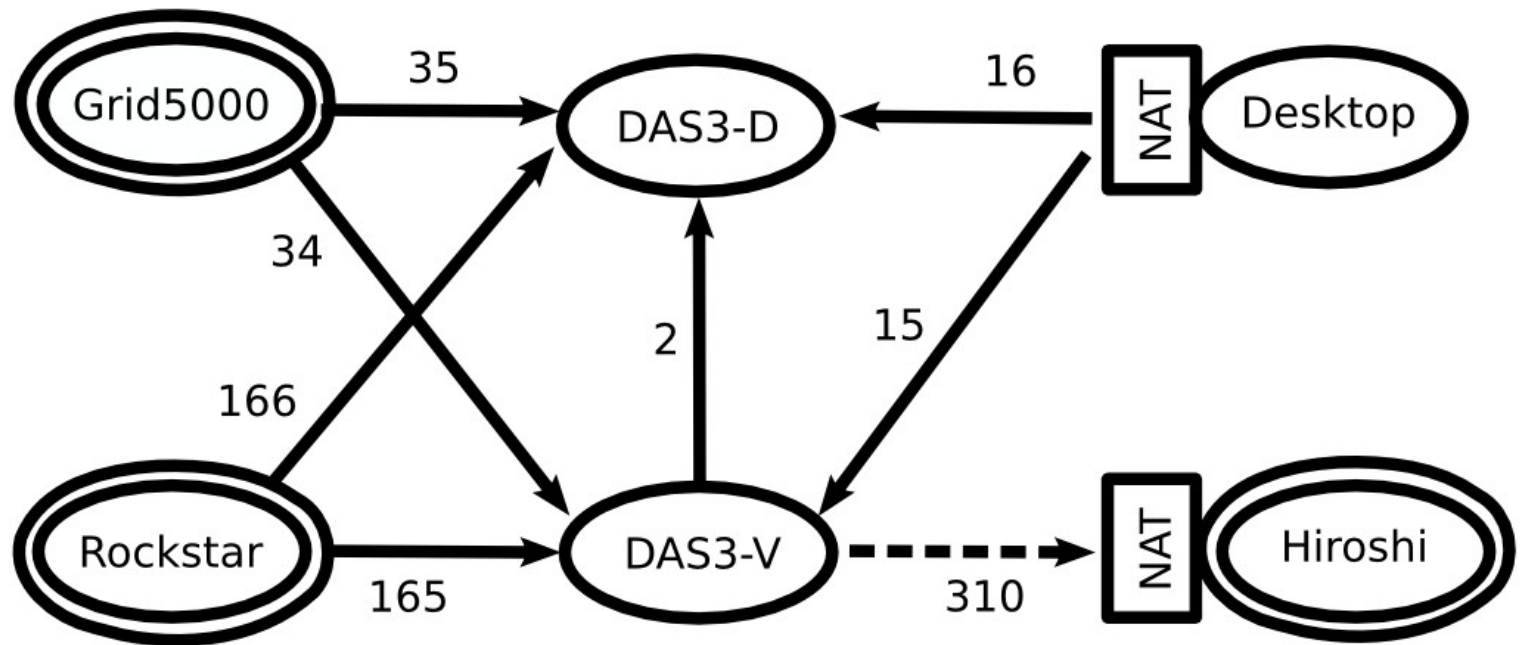
DAS3-V
Amsterdam, NL
MH

DAS3-D
Delft, NL
MH

Grid5000
Sophia Antipolis, France
MH+FW / NR

Hiroshi Cluster
Sydney, Australia
MH+FW+NAT / NR

Hub Network



Evaluation

Table 3: Connection setup time of SmartSockets (time in milliseconds).

<i>Target</i>	<i>Source</i>					
	DAS3-V	DAS3-D	Rockstar	Grid5000	Hiroshi	Desktop
DAS3-V		4.9 ^d (2.4)	332 ^d (166)	68 ^v	595 ^v	33 ^d (17)
DAS3-D	4.9 ^d (2.4)		335 ^d (167)	70 ^v	595 ^v	33 ^d (18)
Rockstar	500 ^r	503 ^r		206 ^v	718 ^v	182 ^v
Grid5000	35 ^v	38 ^v	206 ^v		593 ^v	54 ^v
Hiroshi	630 ^v	603 ^v	750 ^v	670 ^v		640 ^v
Desktop	49 ^r	52 ^r	183 ^v	84 ^v	606 ^v	

Annotations indicate connection style: *d* for direct, *r* for reverse, *s* for splicing, and *v* for routed.

When applicable, the connection setup time of regular sockets is shown between brackets.



Evaluation

Table 4: Roundtrip latency of SmartSockets (time in milliseconds).

<i>Target</i>	<i>Source</i>					
	DAS3-V	DAS3-D	Rockstar	Grid5000	Hiroshi	Desktop
DAS3-V		2.3 (2.3)	166 (166)	56	528	14 (14)
DAS3-D	2.3 (2.3)		167 (167)	57	533	15 (15)
Rockstar	166	167		205	590	195
Grid5000	56	57	205		524	50
Hiroshi	528	529	590	522		539
Desktop	14	15	190	43	522	

When applicable, the roundtrip latency of regular sockets is shown between brackets.

Table 5: Throughput of SmartSockets (in Mbit/second).

<i>Target</i>	<i>Source</i>					
	DAS3-V	DAS3-D	Rockstar	Grid5000	Hiroshi	Desktop
DAS3-V		182 (183)	2.6 (2.5)	2.5	0.25	0.65 (0.65)
DAS3-D	185 (186)		2.6 (2.5)	2.6	0.26	0.65 (0.65)
Rockstar	2.8	2.7		6.9	0.23	0.65
Grid5000	7.6	8.2	2.4		0.20	0.65
Hiroshi	0.73	0.73	0.70	0.73		0.61
Desktop	3.3	3.3	2.2	2.2	0.25	

When applicable, the throughput of regular sockets is shown between brackets.



Other Features

- Reducing connectivity
 - Explicit configuration needed
 - Allows applications to simulate firewalls
 - Can be combined with traffic shaping
 - Simulate a Grid on your cluster!
- Visualization of connections
 - Gives insight into network problems



Future Work

- Alternative protocols
 - UDP instead of TCP
 - Parallel streams
 - Secure connections
- Dynamically switching approach
 - Find a better connection while the socket is in use!
- Hub network scalability
- Multiple hubs per client



Summary

- Communicating on Grids is hard
 - Many connectivity problems occur
 - Takes a lot of work to find the problems and work around them
- SmartSockets reduces this into a single problem:
 - **How to set up a spanning tree of hubs**
- The rest is done automatically!

