Slingshot: Deploying Stateful Services in Wireless Hotspots

Ya-Yunn Su

Jason Flinn

University of Michigan

Motivation

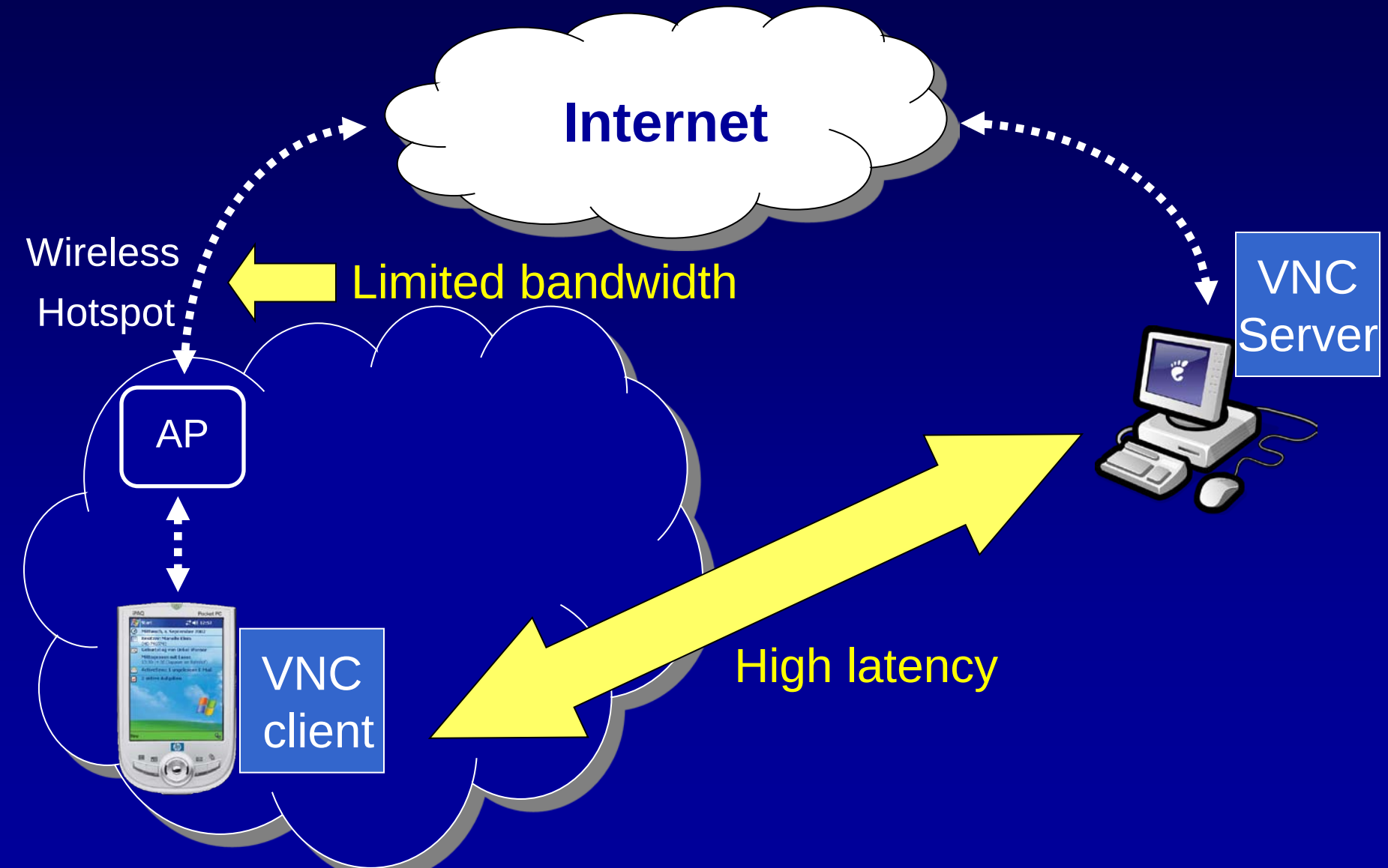
Performance

Mobile Computer

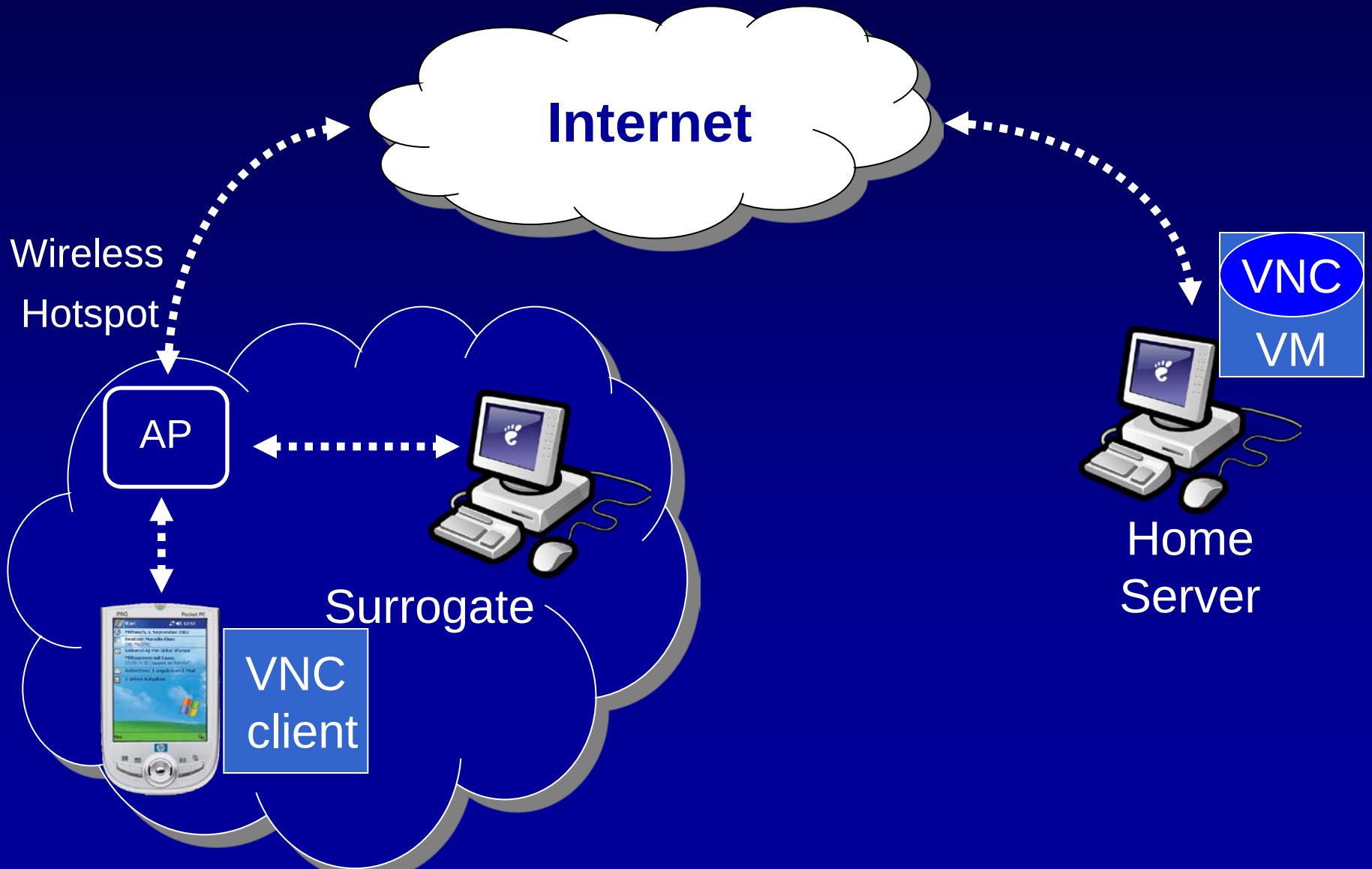
Portability

- Portable: take everywhere
  - Easy to carry and less obtrusive
  - Limited in resources
- Performance: run demanding applications
  - More processing power and storage capacity
  - Bulkier and heavier

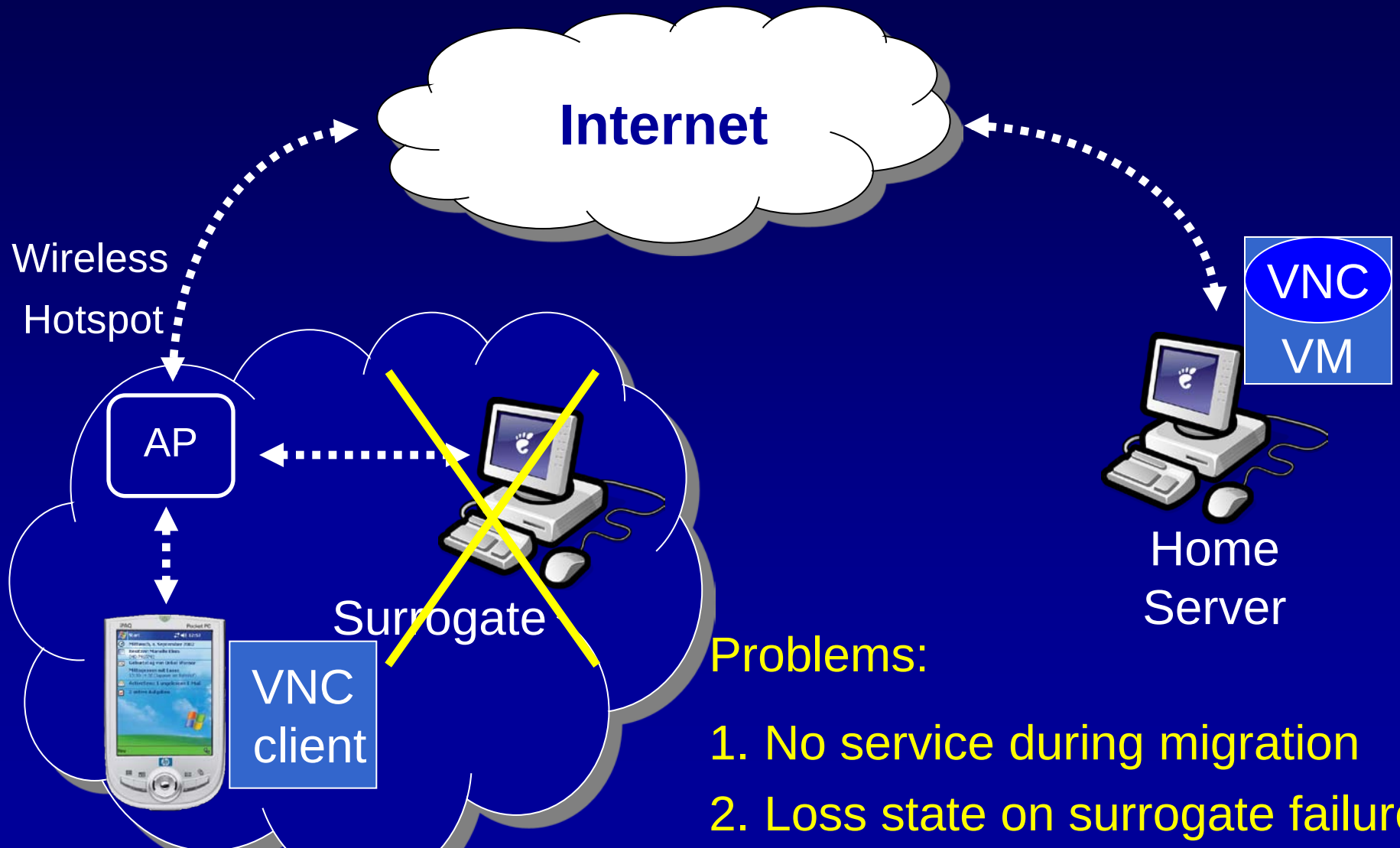
Remote Execution



Cyber Foraging



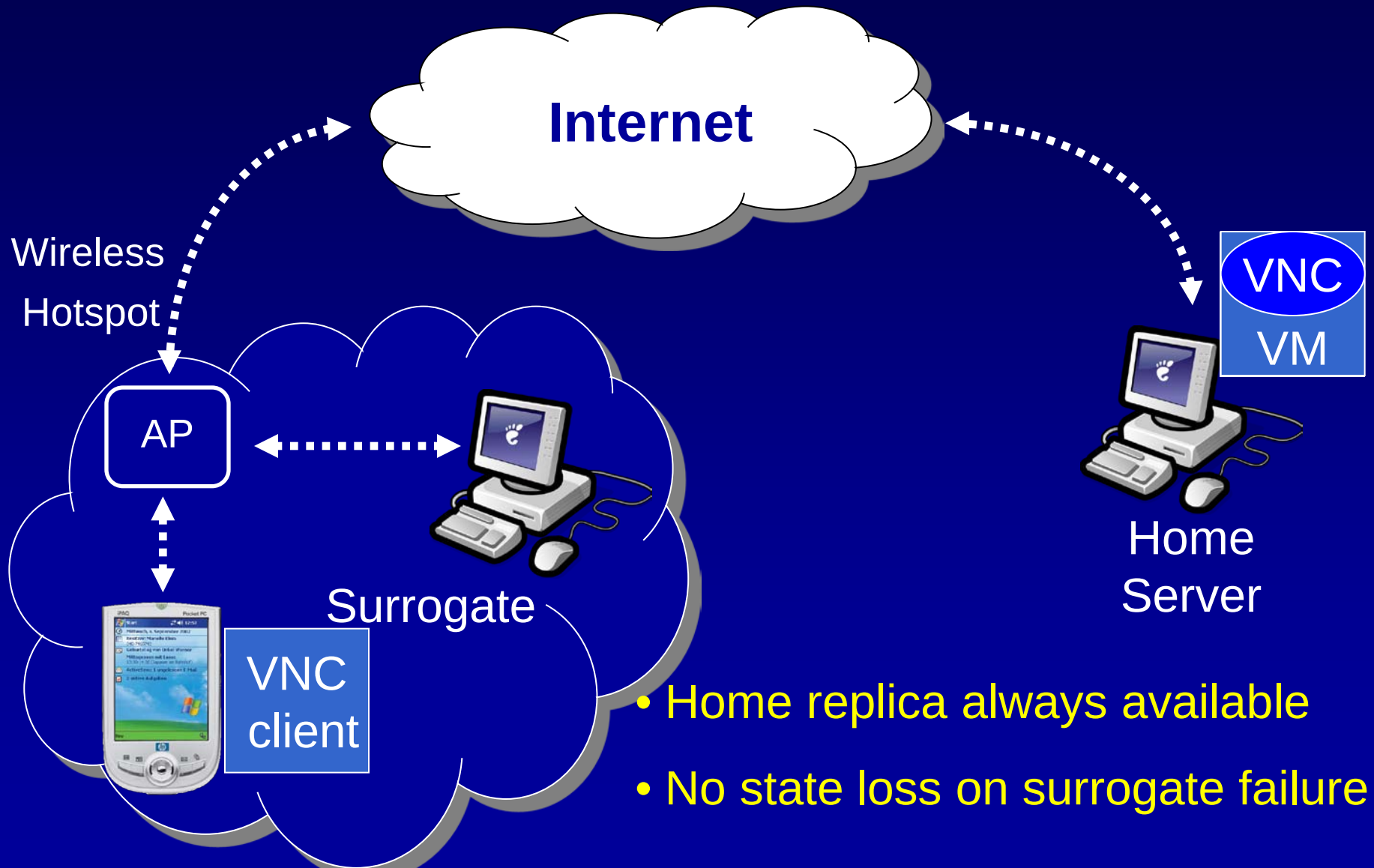
Migrating Remote Services



Problems:

1. No service during migration
2. Loss state on surrogate failure

Slingshot: Replicate Services



- Home replica always available
- No state loss on surrogate failure

Ease of Management

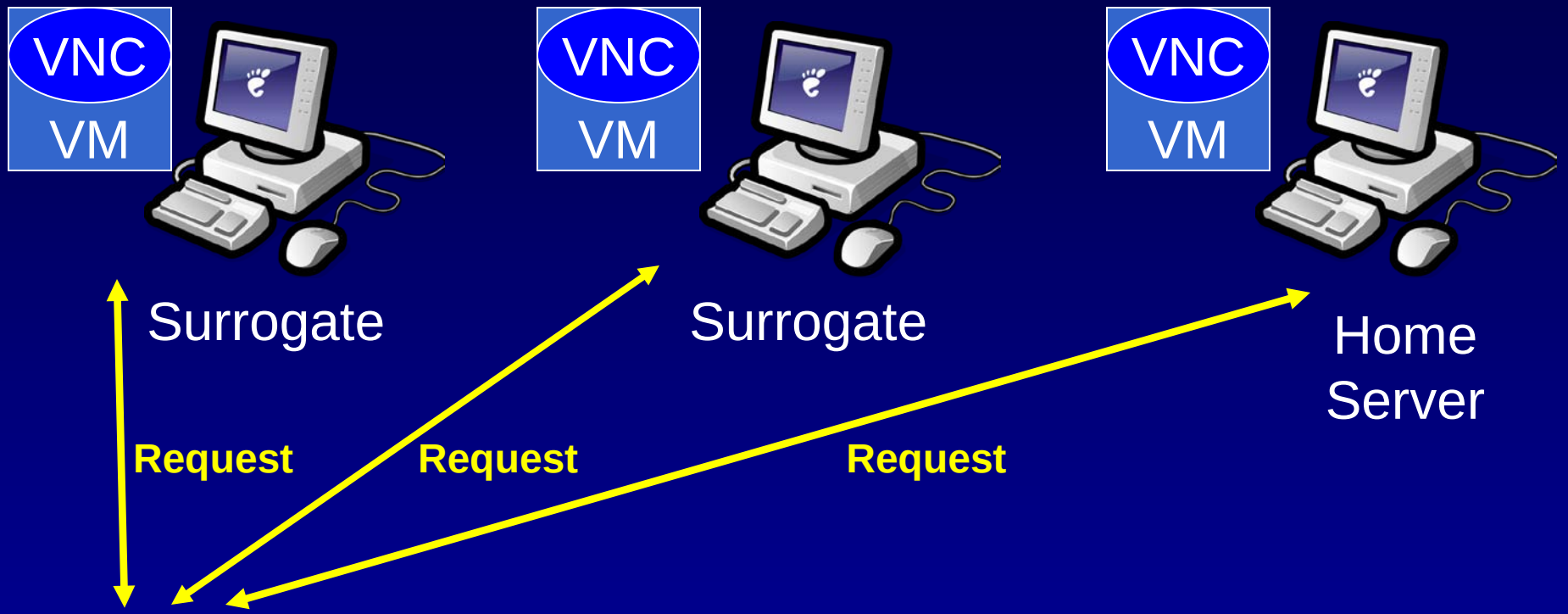
- Surrogates should be appliances

- Slingshot
  - Minimizes the surrogate computing base
  - Uses a heavyweight virtual machine
  - Places no hard state on surrogates

Outline

- Motivation
- Implementation
- Evaluation
- Related Work
- Conclusion

Slingshot Overview



VNC proxy

VNC client

- Slingshot applications
  - Remote desktop: VNC
  - Speech recognition: IBM ViaVoice

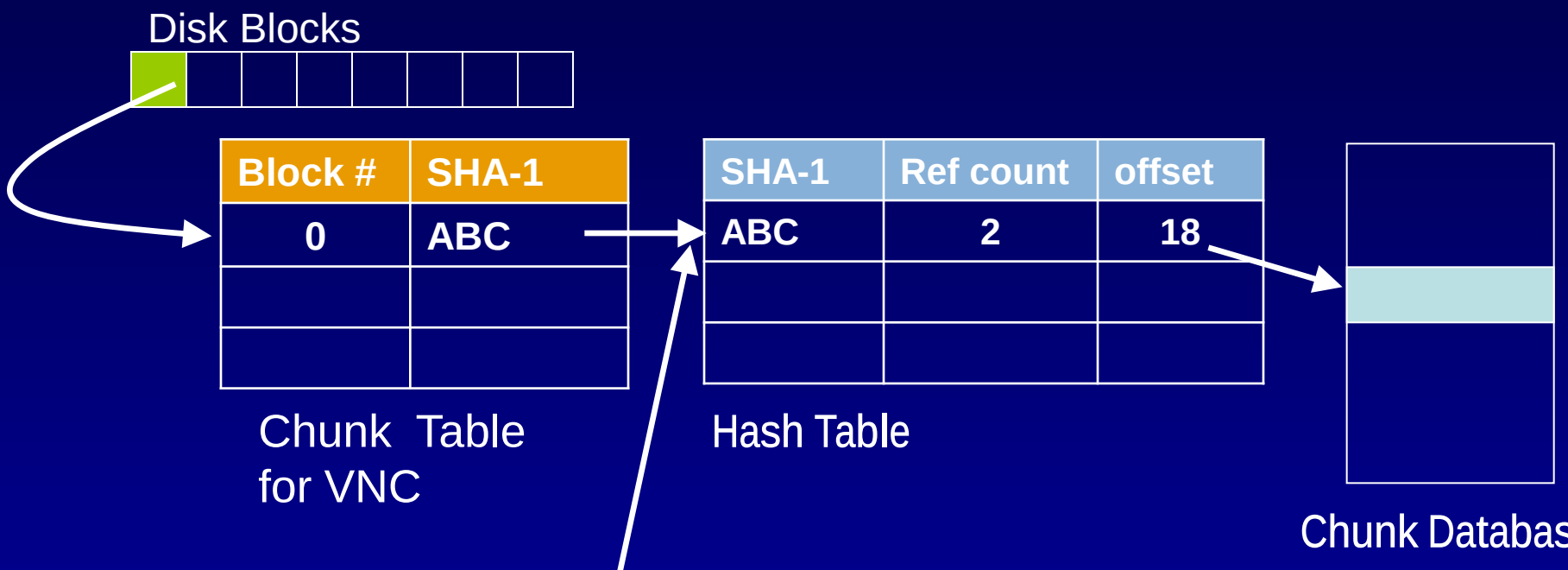
State of a Remote Service

- The virtual machine state contains:

  Volatile state: memory image and registers
  - Unique to each service
  - Compressed and stored as individual files

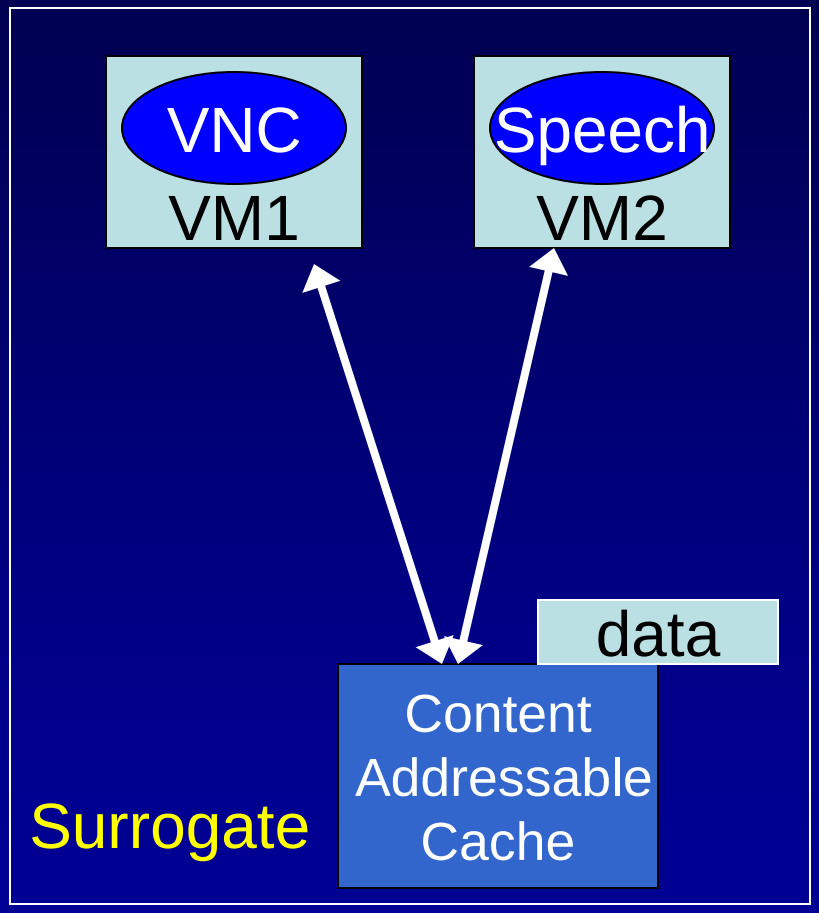
  Persistent state: virtual disk image
  - Large: ex. 4 GB for our VNC service
  - Stored in content addressable database

Content Addressable Database



- This runs on the home server
- Built upon Internet Suspend Resume *[Kozuch02]*

Surrogate



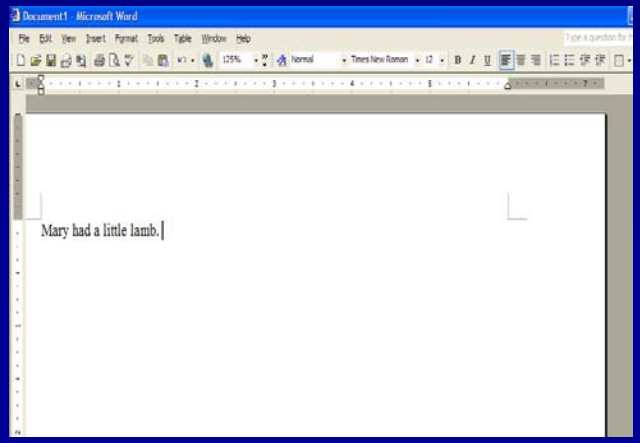
Content
Addressable
Database

Home Server

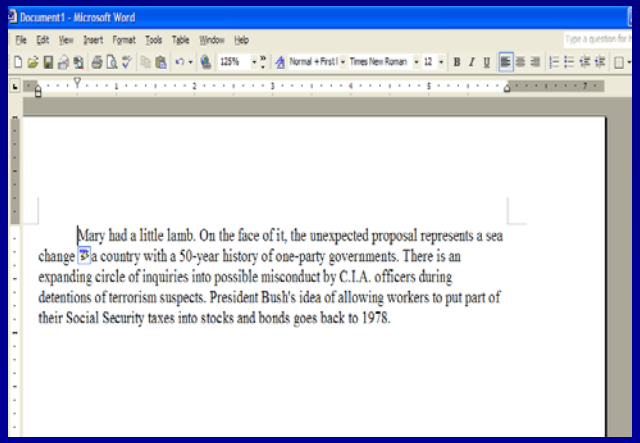
- Persistent state can be
  - Fetched on demand
  - Shared between applications

Creating a Replica

- Big idea: recreate current state from
  - Checkpoint on the home server
  - Event log on the client
- Application level determinism
- Example:



Keystroke
Keystroke
Keystroke
Mouse movement



Checkpoint + Event Log = Current State

Instantiate a Replica

1. Checkpoint

VNC
VM

VNC
VM

2. Transfer volatile state and chunk table

SLOW!

Surrogate

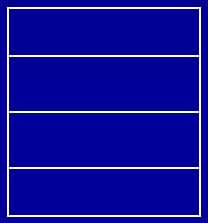
Home server

3. Replay event log

VNC proxy

VNC client

2. Log requests



Leveraging Portable Storage

Surrogate

Home server

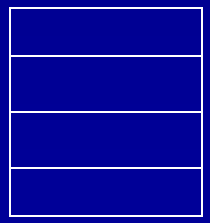
1. Transfer volatile state, chunk table and hash table

2. Replay event log

VNC proxy

VNC client

1. Log requests



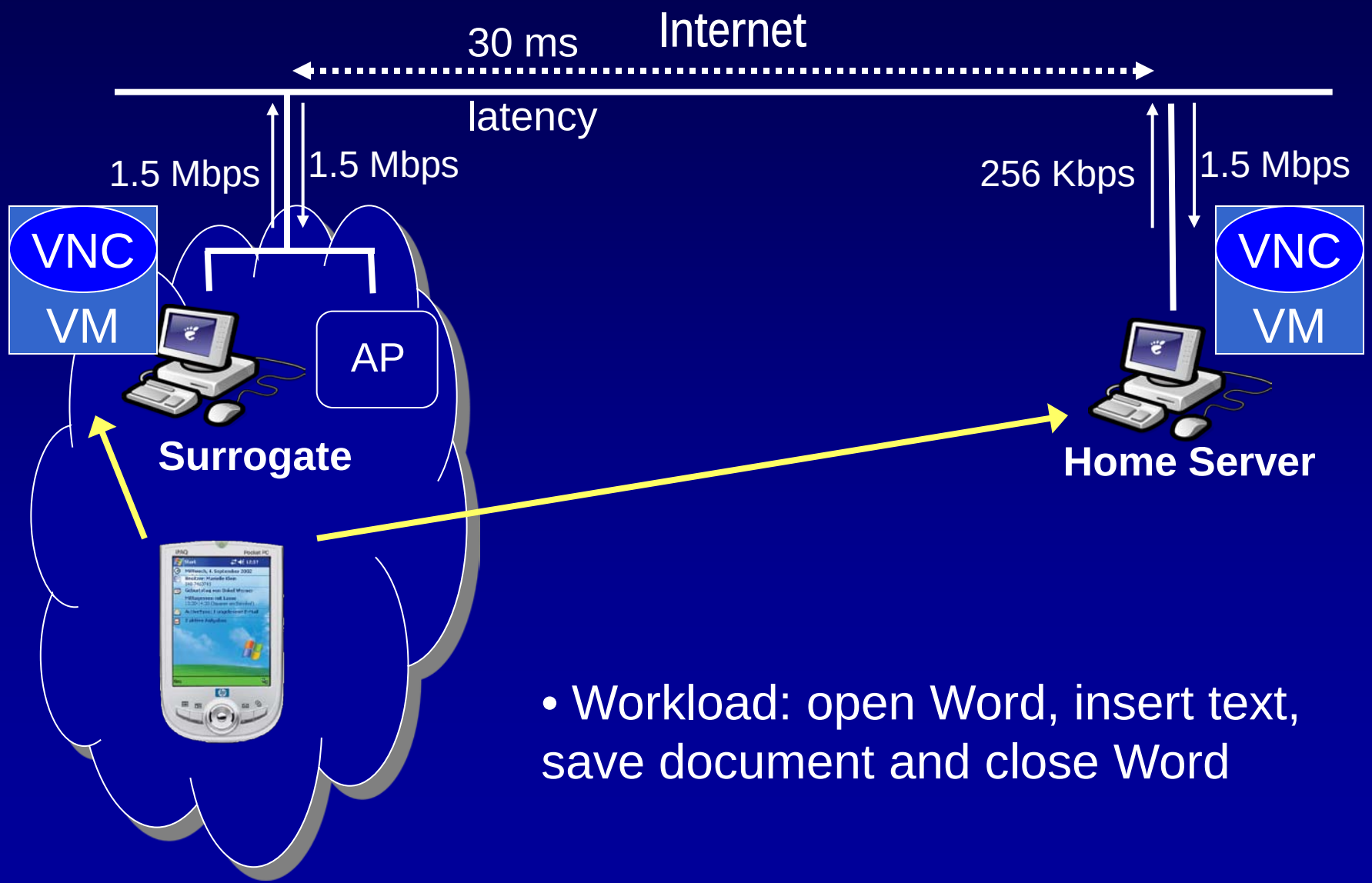
Portable storage carries:

- **Volatile state**
- **Content addressable database**

Outline

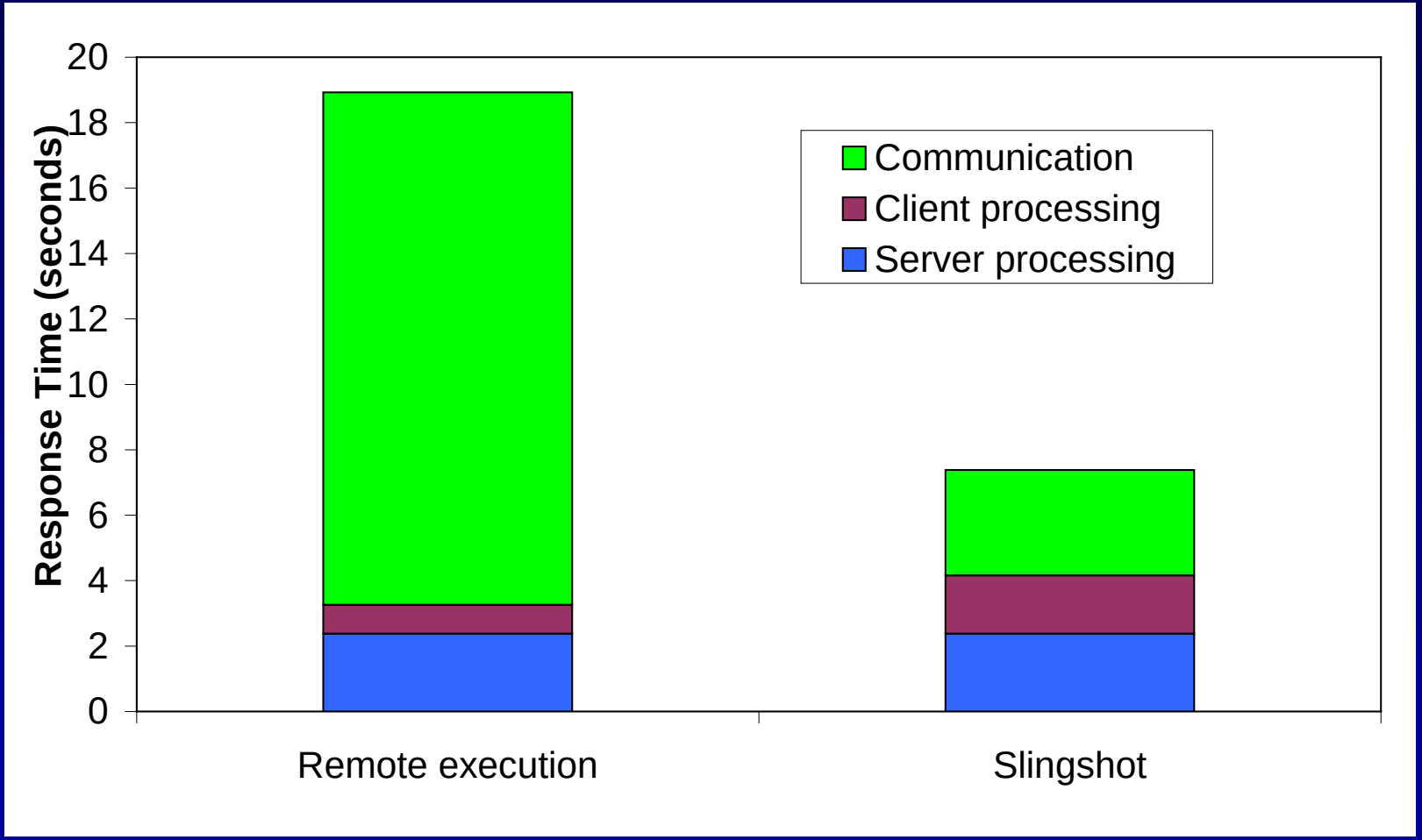
- Motivation
- Design Principles
- Implementation
- Evaluation
- Related Work
- Conclusion

Network Topology



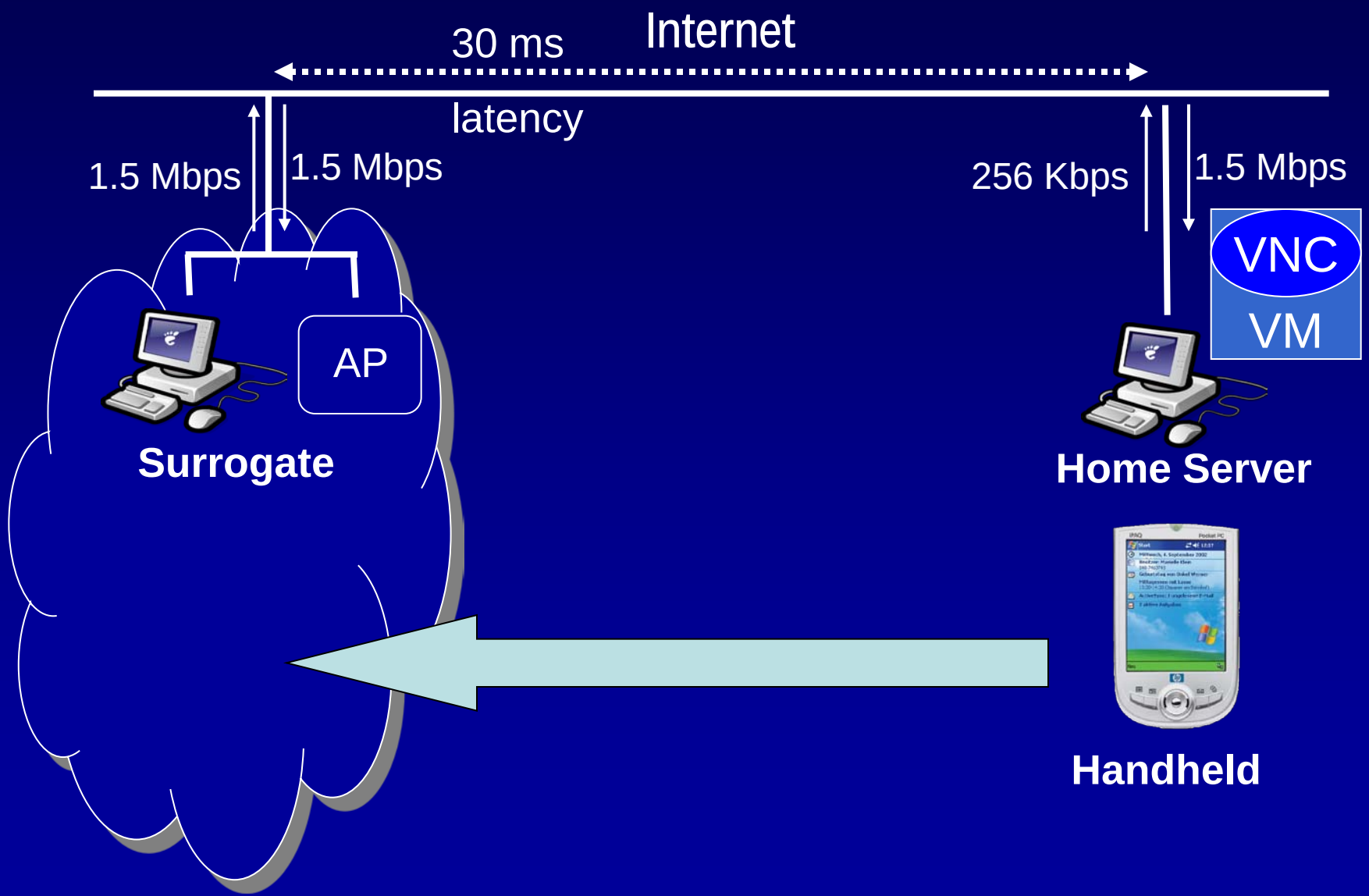
- Workload: open Word, insert text, save document and close Word

Benefit of Slingshot

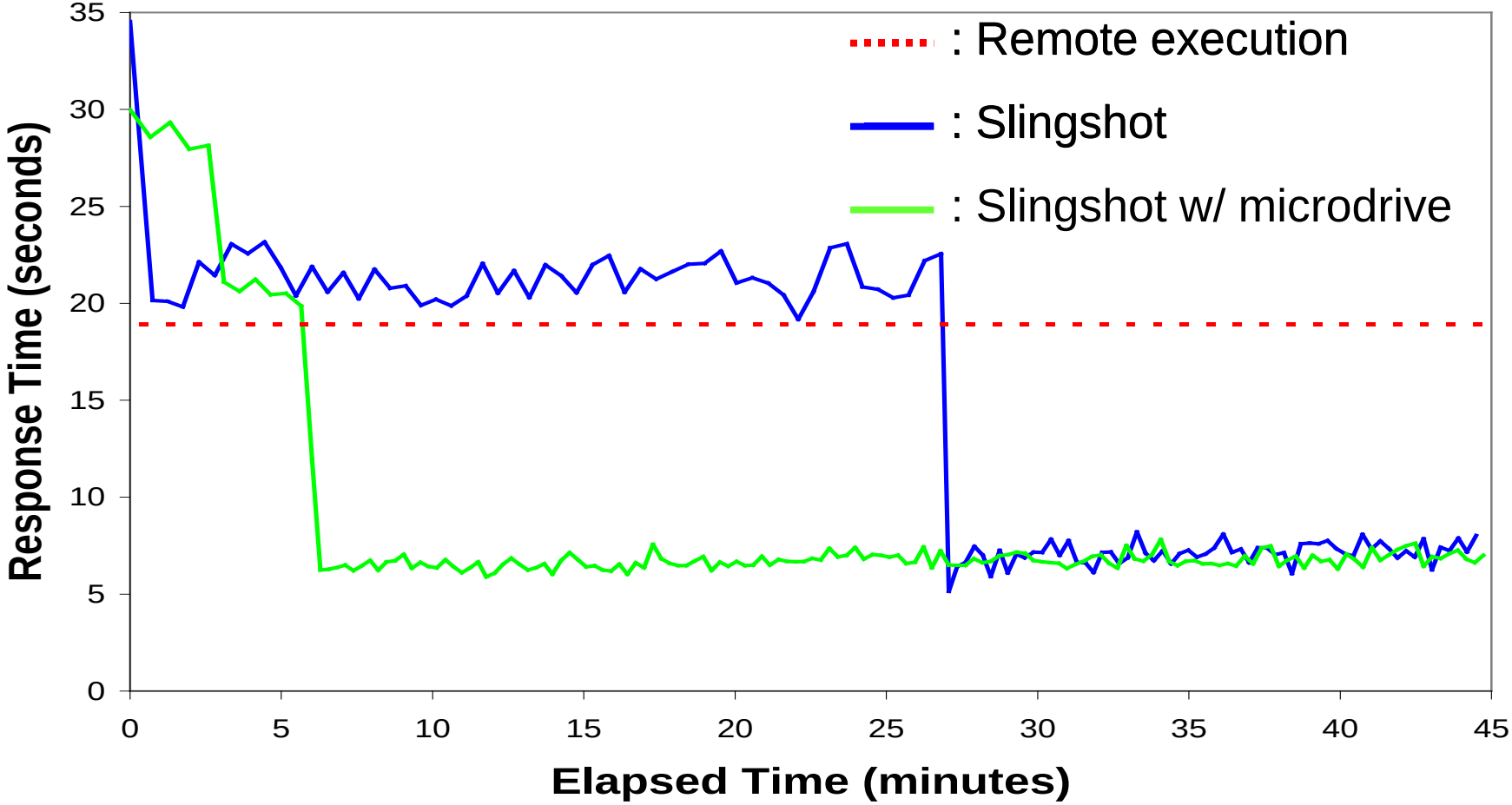


- Slingshot: 2.6 times faster than remote execution

Network Topology

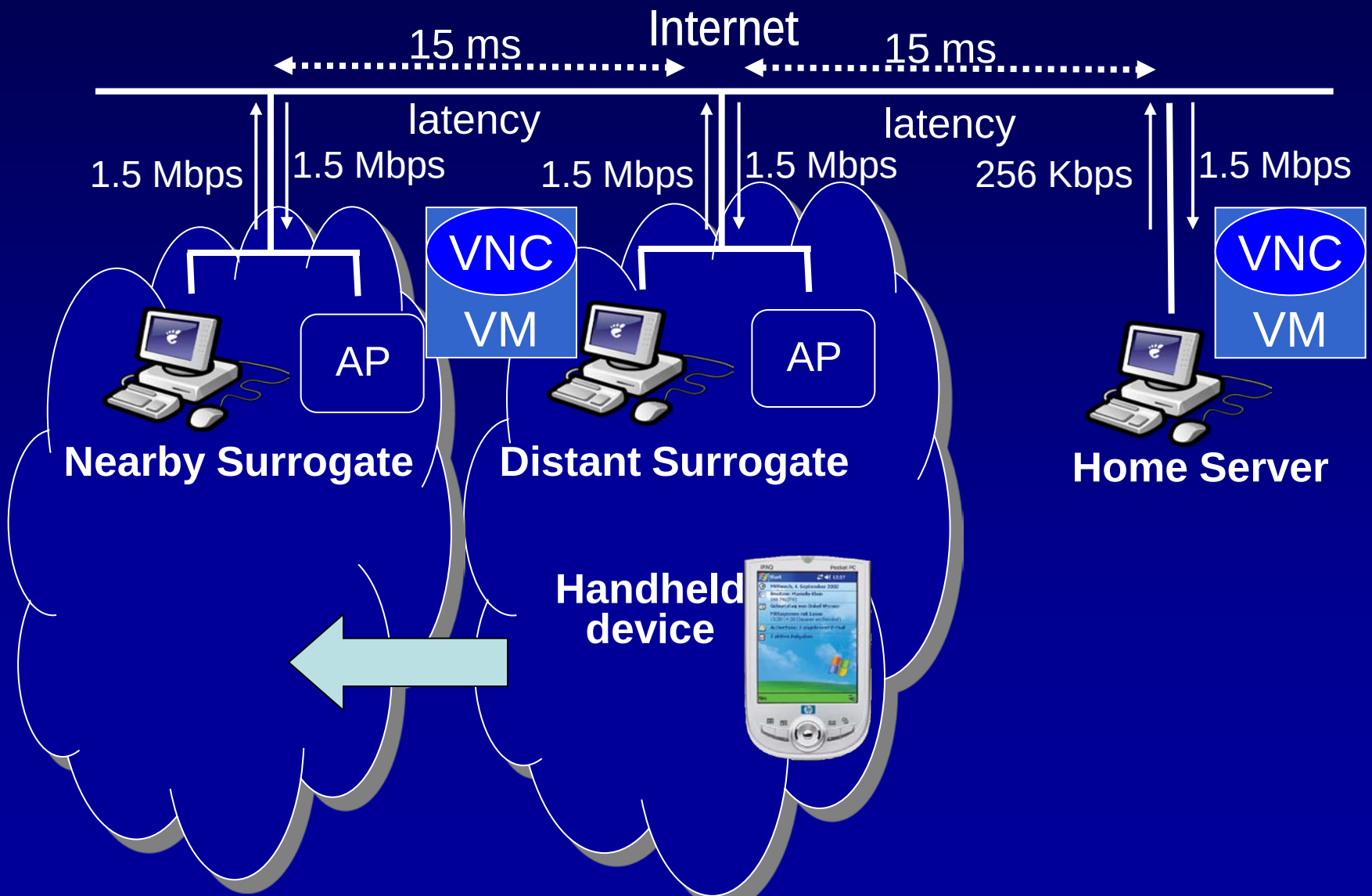


Instantiating the First Replica

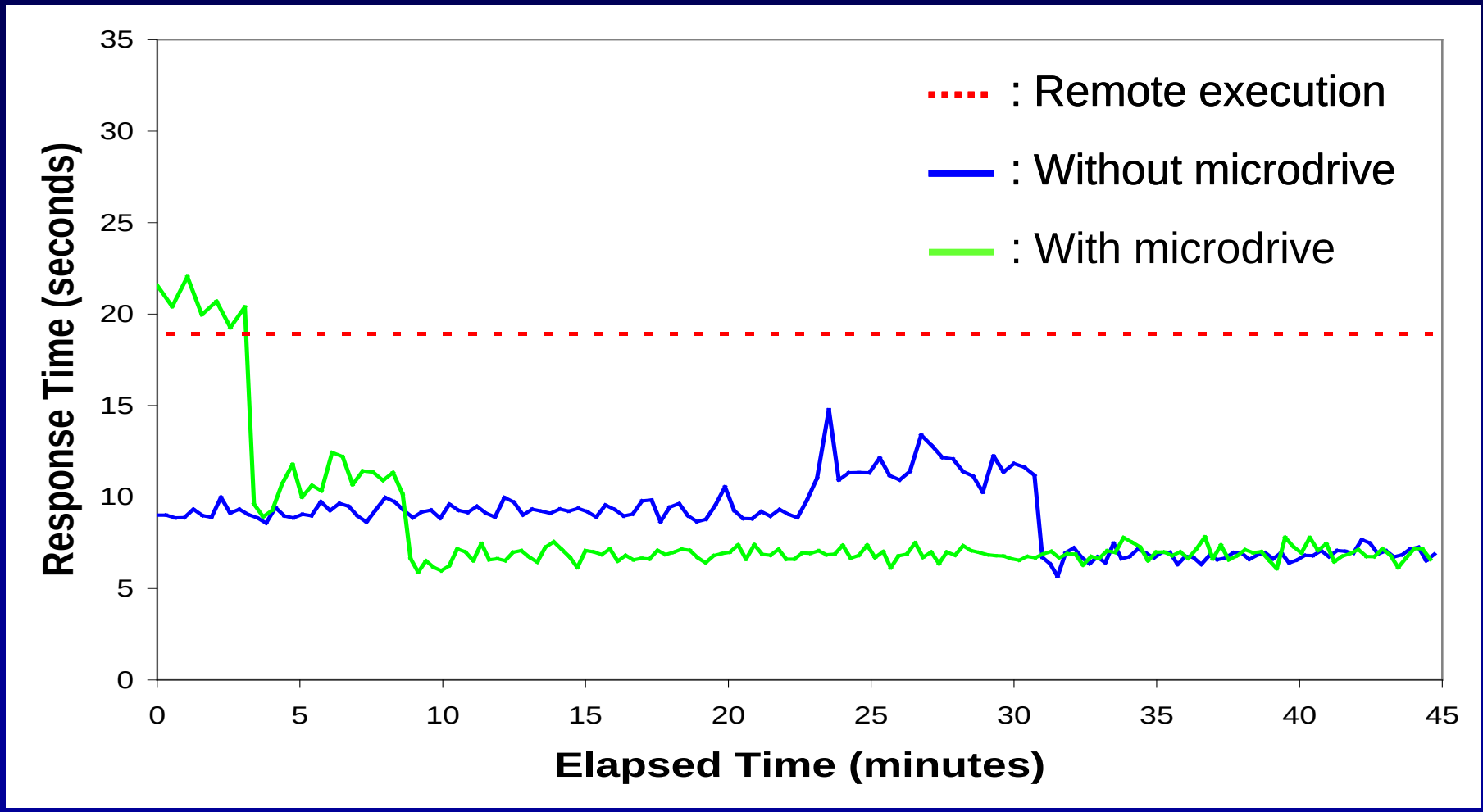


- Slingshot executes 2.6 times faster than remote execution

Network Topology



Instantiating Another Replica



Related Work

- Cyber foraging *[Balan03, Goyal04]*
  - Support user mobility and stateful services
- Virtual machine/process migration
  [Sapuntzakis02, Kozuch02, Tolia03]
  - Apply the same optimization techniques
  - Replicate VM to service mobile computers
- Replay at different level
  [Dunlap02, Bressoud95, Rodrigues01, Brown02]
  - Enforce determinism at the application level

Conclusion

- Slingshot
  - Is 2.6x faster than remote execution
  - Hides surrogate failure
  - Minimizes surrogate maintenance cost
- Questions?