

INTELLIGENT VIRTUAL MACHINE STATE PREFETCHING

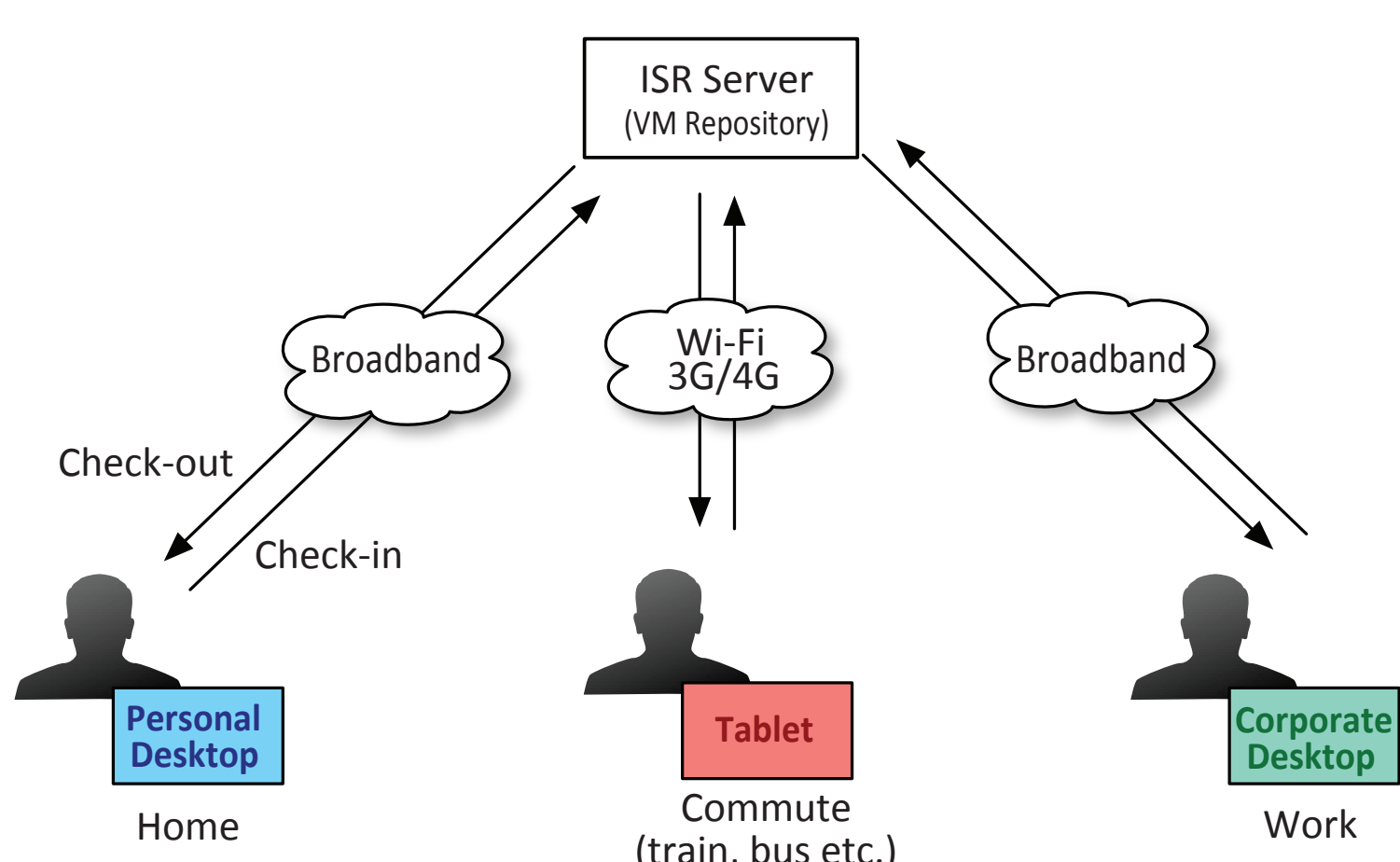
Yoshihisa Abe and Mahadev Satyanarayanan (CMU)

MOTIVATION

- Post-copy VM migration with smart state prefetching is deemed desirable in diverse settings
 - Cloudlet handoffs
 - Migration over WANs
 - VM use with minimal local state for security
- Efficiency of VM state transfer matters
 - Minimizing wait for users physically moving
 - Minimizing disruption of VM execution due to faulted-in state
 - Feasibility over low-bandwidth networks
- Make VM state prefetching smart by exploiting VM state semantics
 - Guest cooperation in characterizing VM memory regions, VMM-level inspection of address spaces etc.

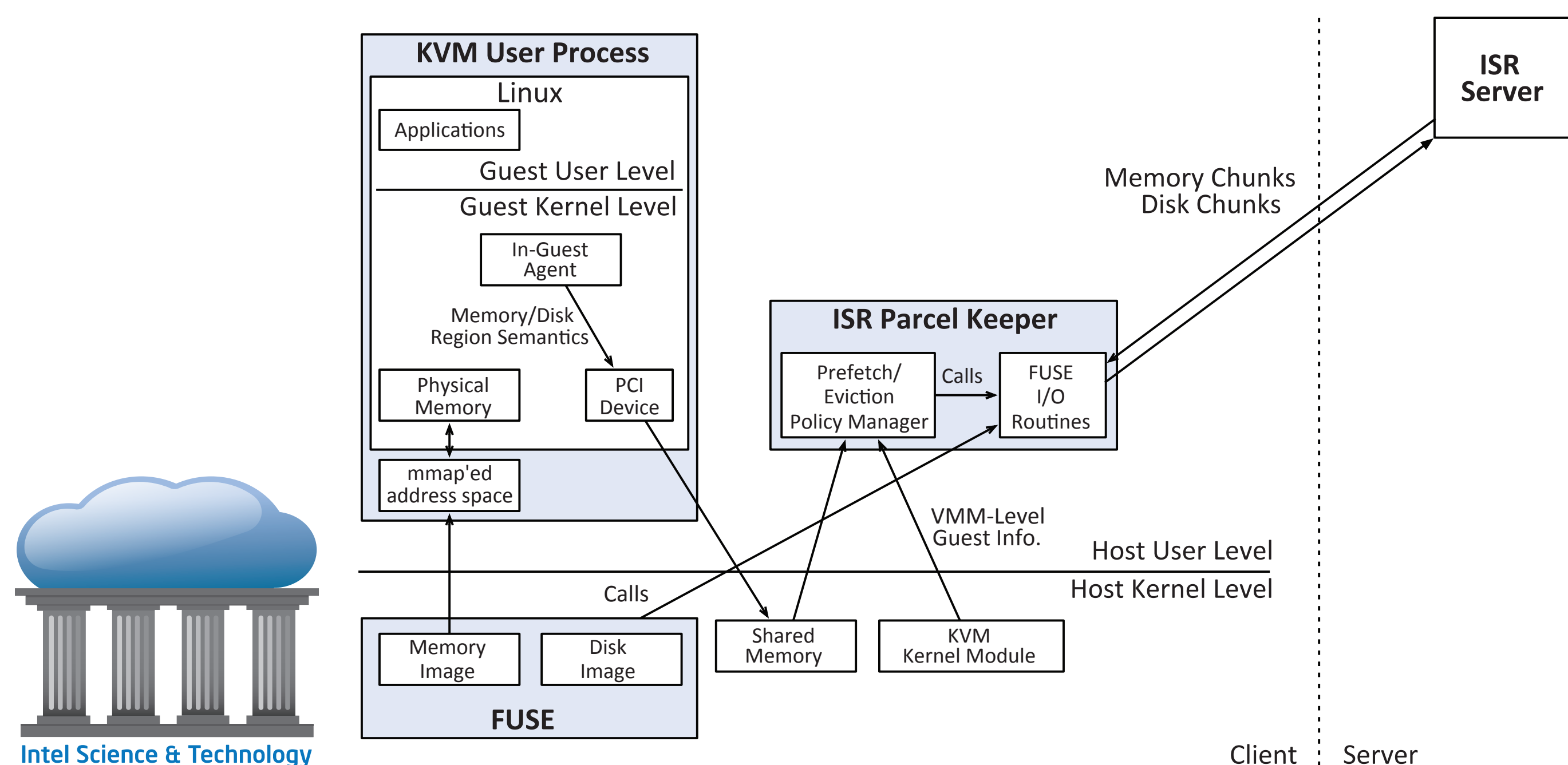
VM STREAMING OVER WANs

- Enable universal access to single computing environment from multiple physical locations/devices
 - Free users from setting up and updating each device separately
 - Both computation and data made accessible by encapsulation in VM
 - Allow taking advantage of available devices at different locations
 - Desktop with large display at work, laptop/tablet while commuting etc.
- Let user start using VM with minimal wait
 - Launch VM with partial local state and prefetch the rest as necessary



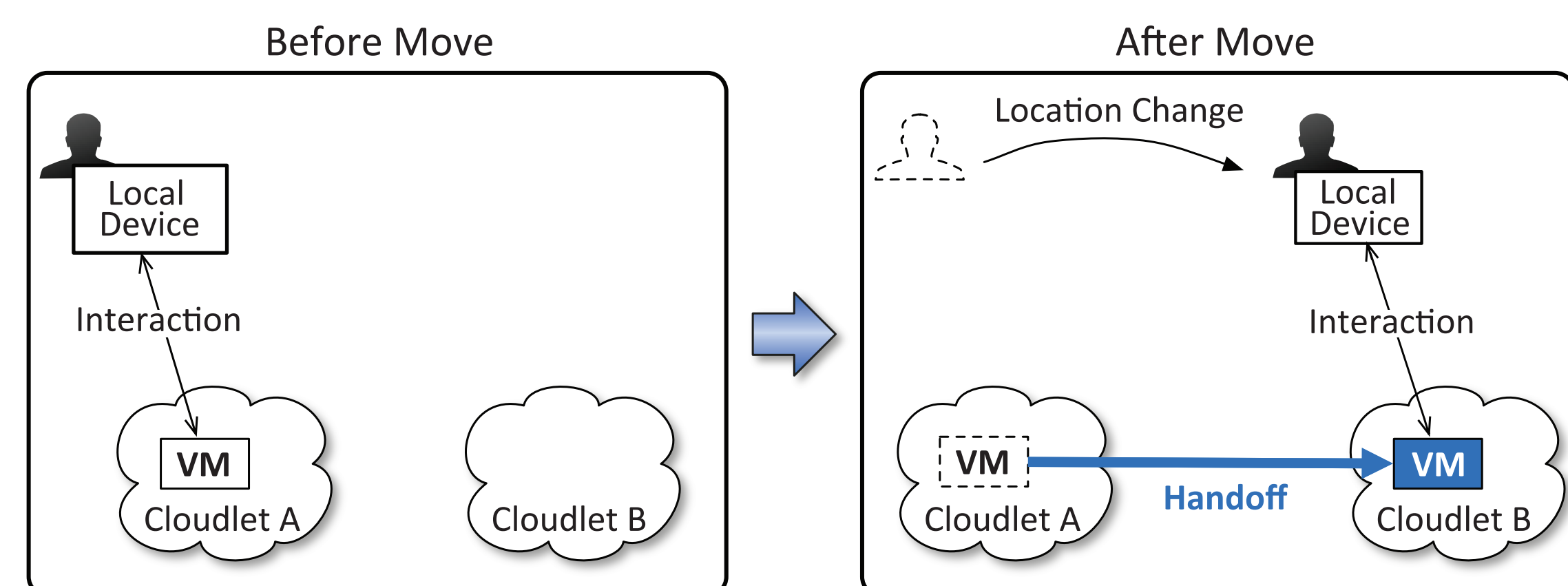
SYSTEM ARCHITECTURE

- Modified version of Internet Suspend/Resume (ISR)
 - Hypervisor: KVM, Guest: Linux (Ubuntu)
 - Asynchronous memory/disk state transfer between client and server while VM executes
 - In-guest kernel agent and KVM kernel module provide VM state semantics



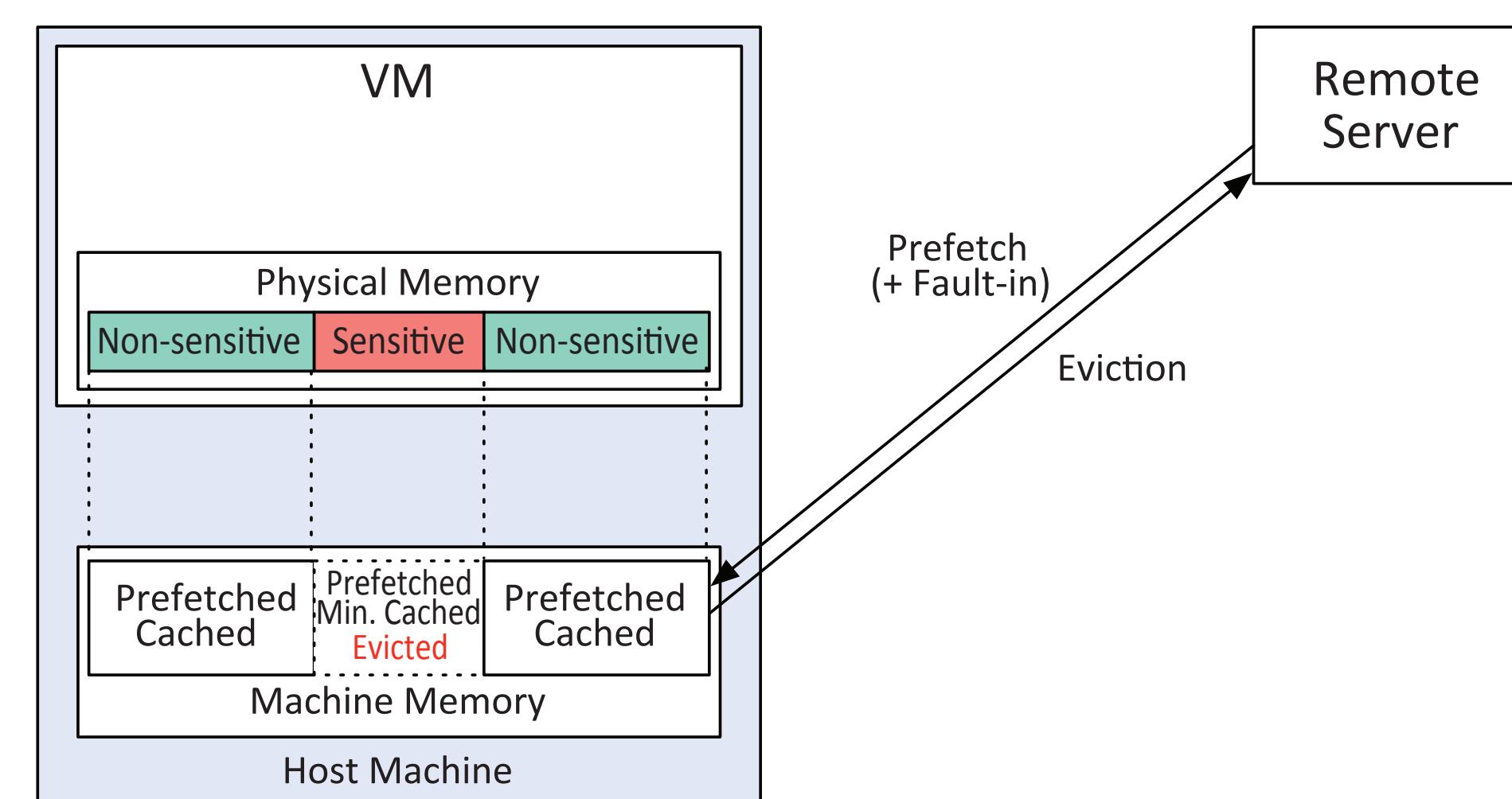
MOBILITY-INDUCED CLOUDLET HANDOFFS

- Transfer VM between cloudlets upon user's location changes
 - VM runs at cloudlet location closest to user
 - Support interaction better between user's device and VM
 - Transfer made seamless by efficient, dynamic state transfer
- Particularly useful in military scenarios



SECURE VM USE WITH STATE EVICTION

- Device loss is becoming a major concern
 - Employee's laptop lost or stolen leaks confidential corporate information
 - Lost smartphone threatens user's privacy
- Minimize risk of exposing sensitive data by constant eviction from local device
 - Evicted data is re-fetched from remote server as needed



WORK IN PROGRESS

Current platform supporting:

- Chunked memory (as well as disk) image for KVM with a fetching mechanism
- Semantic information flows from guest Linux and KVM kernel module
 - Both take advantage of Linux kernel tracing facility

Implementation in progress:

- Client-to-server state transfer
 - Support for sensitive data eviction
 - Invalidate memory mappings
 - Zero out caches at multiple levels etc.
- Prefetch/Eviction Policy Manager
 - Decide what parts of VM state to prefetch or evict
 - Exploits guest cooperation and VMM-level instrumentation for hints

