ROMA: A Middleware Framework for Seamless Handover

*Adrian Popescu, **David Erman**, Karel de Vogeleer, Alexandru Popescu, Markus Fiedler*

www.bth.se

BLEKINGE INSTITUTE OF TECHNOLOGY

euronf

NETWORK OF EXCELLENCE

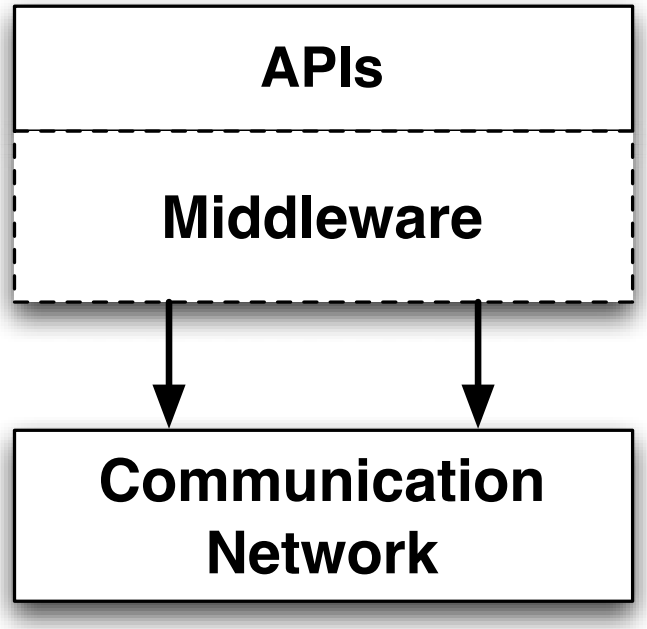
BLEKINGE TEKNISKA HÖGSKOLA · BTH ·

Situation today

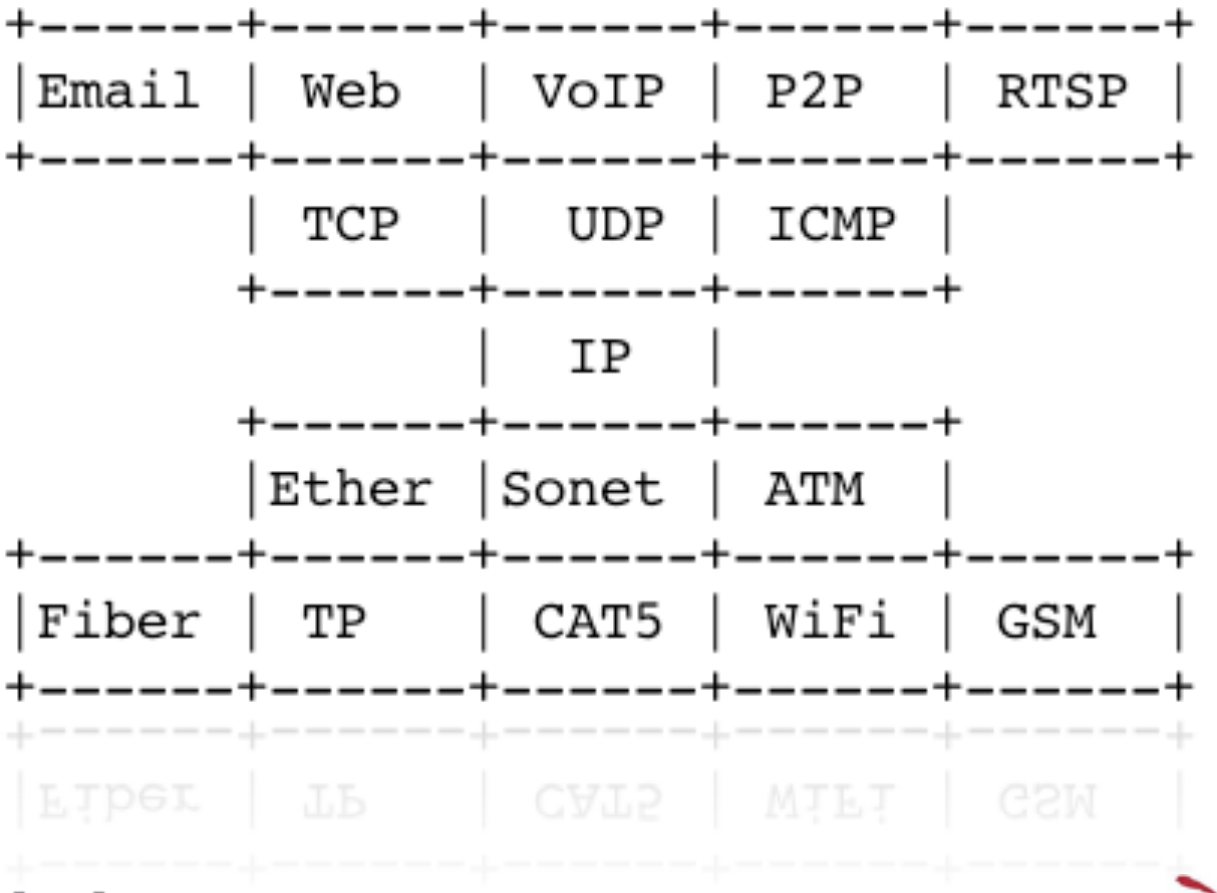
- Device heterogeneity
- Access network heterogeneity
- Application heterogeneity
- Trend towards complete connectivity everywhere
- Solutions?
  - IP: “One protocol to rule them all, one protocol to find them, one protocol to bring them all and in the socket bind them”
- Middlewares to abstract the complexities of connectivity

A very simple middleware abstraction

- Two main tasks:
  - Export API for applications
  - Interface one or more communication networks using protocols
- But the devil is in the details

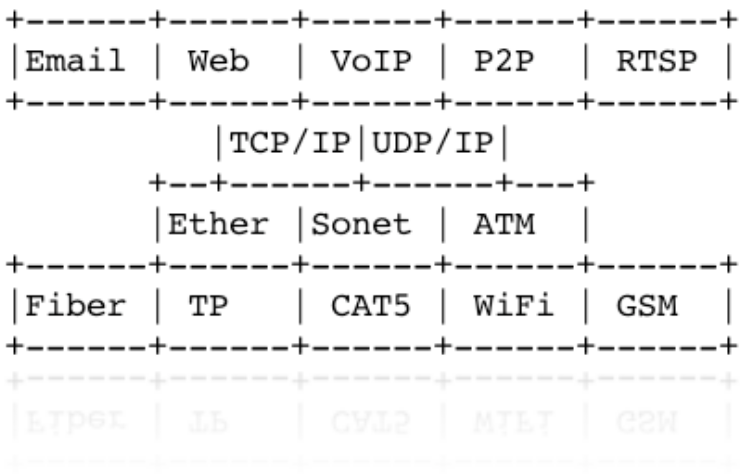


But, where did the hourglass go?



NAT made it fat!

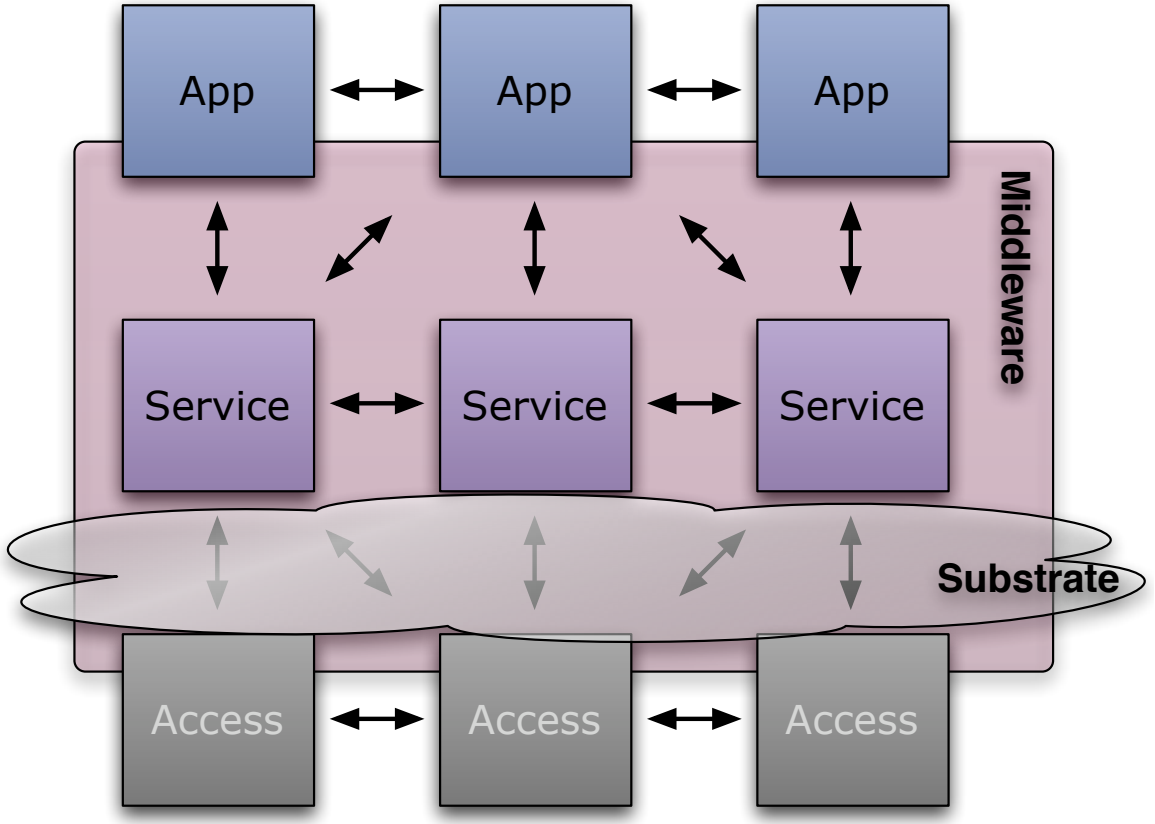
- What about the top of the hourglass?
- Also got fatter
  - More applications
  - More Middlewares!
- The hourglass is getting top-heavy?



Handling complexity?

- Current trend in MW seems to be the classical CS solution to every problem:
  - “Another level of indirection/abstraction.”
- Meta-middlewares, a.k.a. abstract platforms, are appearing
  - SMILE – Simple Middleware-independent Layer
  - ROMA – A Middleware Framework for Robust Mobile Applications
  - MDA – Model-Driven Architecture
- The end result of this?
  - We have a set of abstractions that abstract abstractions, which in turn abstract other abstractions, that abstract…
  - It’s easier to view MW networking as composed of *functional blocks*, rather than a layered architecture,

Functional view



The ROMA Architecture

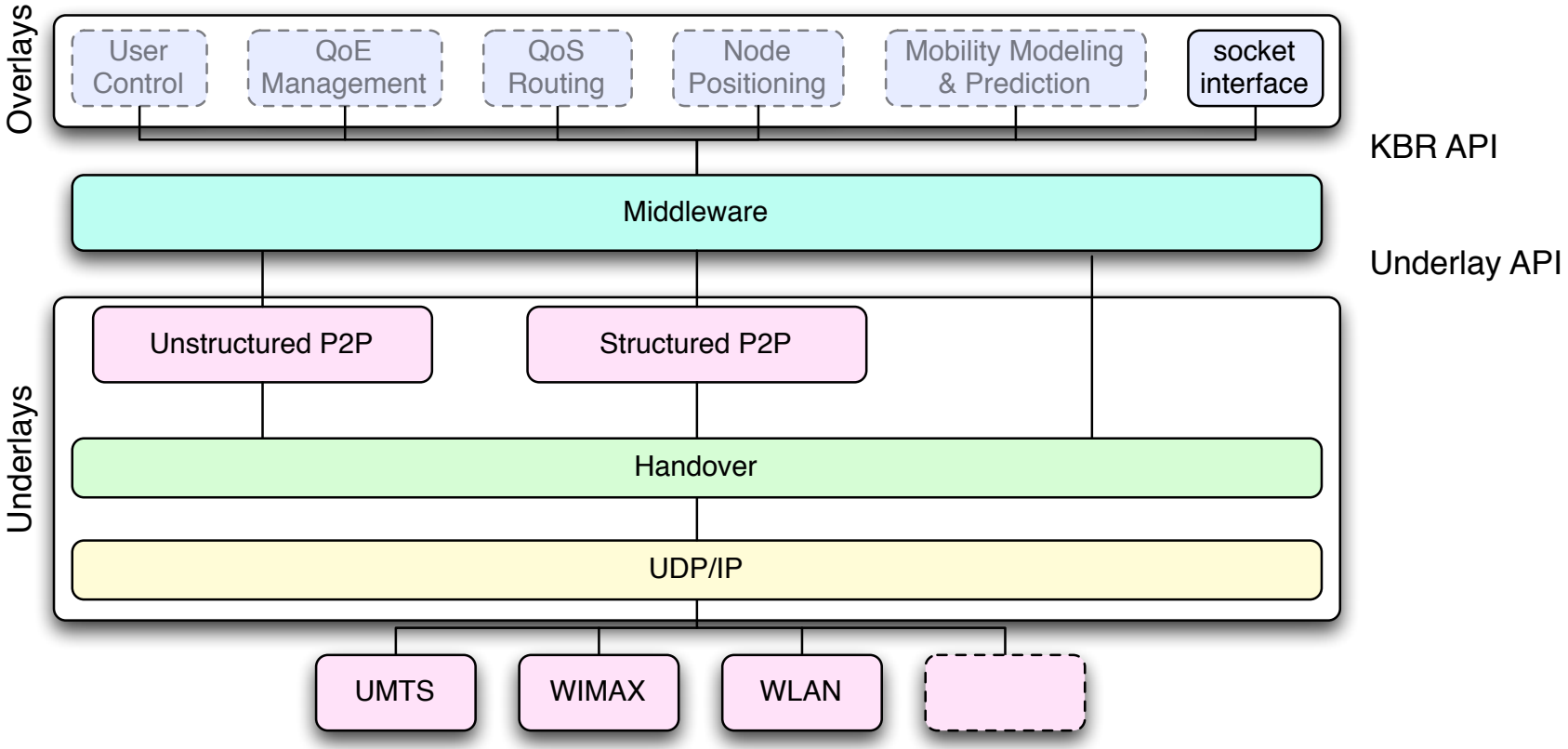
The Architecture in Short

- Application-layer architecture for abstracting network connectivity(ies)
- Originally intended for performance studies of P2P overlays
  - Extended to multi-interface handover
- Manages multiple interfaces and applications
- Very simple API for application developers
- *Fairly* simple API for interfacing (shimming) underlying transport technologies

Goals and requirements

- Goal: Develop testbed for development, testing, evaluation and performance analysis of different solutions for user-centric mobility
  - ROMA Middleware is the software representation of this testbed
- Requirements:
  - Minimal changes to existing applications
  - Support multiple platforms
  - Hide implementation style (i.e., local, client–server & P2P)
  - Provide network-agnostic transport enabler

The ROMA Architecture



Overlays and Underlays

- An *Overlay* provides an application-level routed and controlled network over another network
- An *Underlay* is an abstraction of a transport substrate
- A *substrate* is any network that can provide a packet transport function
  - Other overlays
  - “Normal” IP

API Design

- ROMA consists of two distinct APIs
- “Upper” API based on the Key-Based Routing (KBR) API by Dabek, et al.
  - Provides 160-bit addressing scheme and small set of routing primitives
- “Lower” API to abstract underlays
  - Shim layers for each underlay
- Main challenges:
  - Address translations
  - Shim implementations

Current implementation status

- Basic forwarding implemented
  - IP shim
- Asynchronous event handling
  - Linux epoll() -> boost asio library
- Open issues
  - Kernel-level IF switcher (for handover)
  - Flow multiplexing, due to multiple IFs
  - ROMA middleware is app-monolithic, i.e., each application gets its own copy of ROMA.

What next?

- Implement vertical HO and compare to existing tunneling solution developed at BTH
- Deploy for large-scale tests on PlanetLab

Thank you for your time!

<?>

www.bth.se

BLEKINGE INSTITUTE OF TECHNOLOGY

euronf

NETWORK OF EXCELLENCE

BLEKINGE TEKNISKA HÖGSKOLA · BTH ·