

Mobile Code Offloading: Should it be a Local Decision or Global Inference?

Background

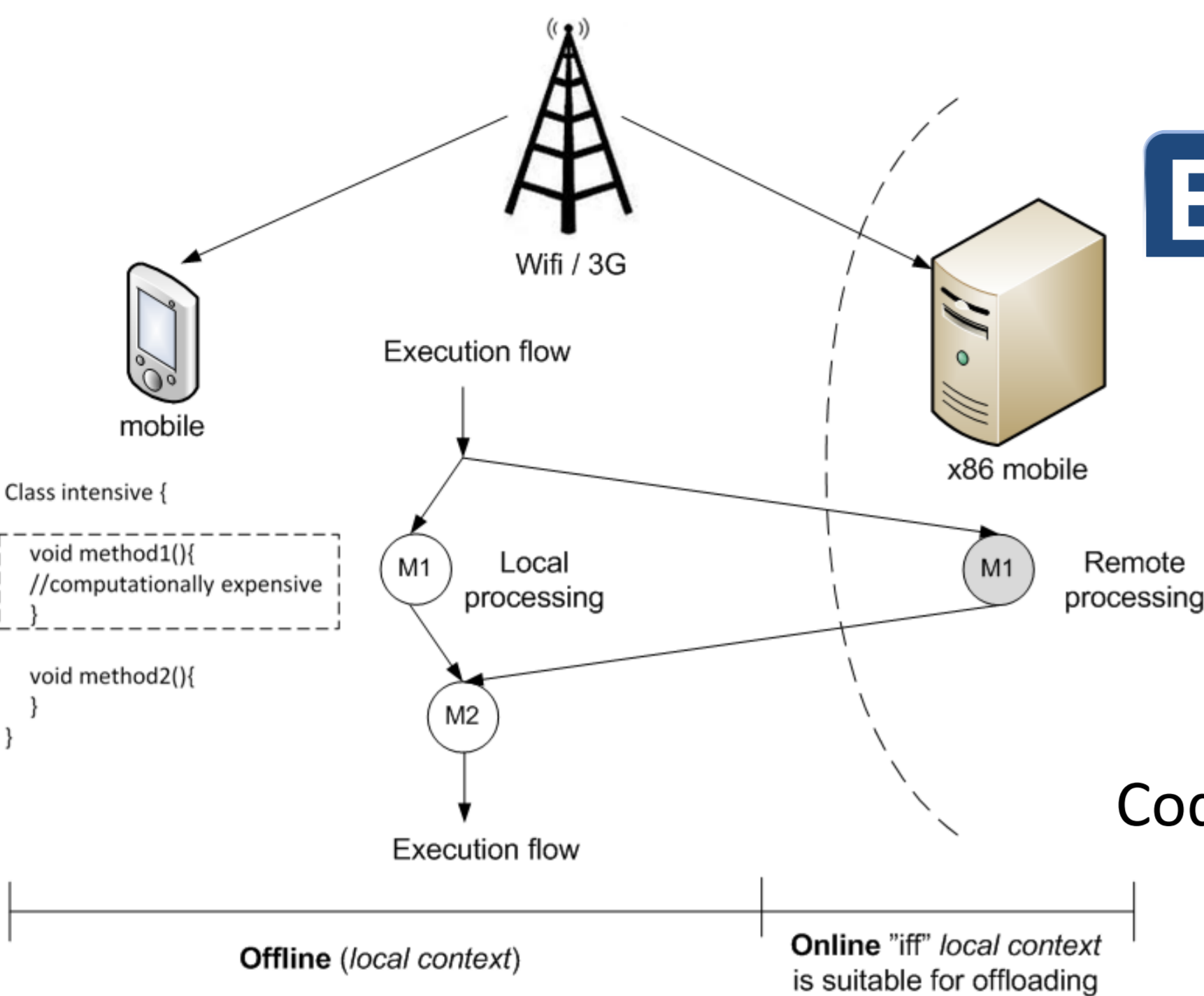
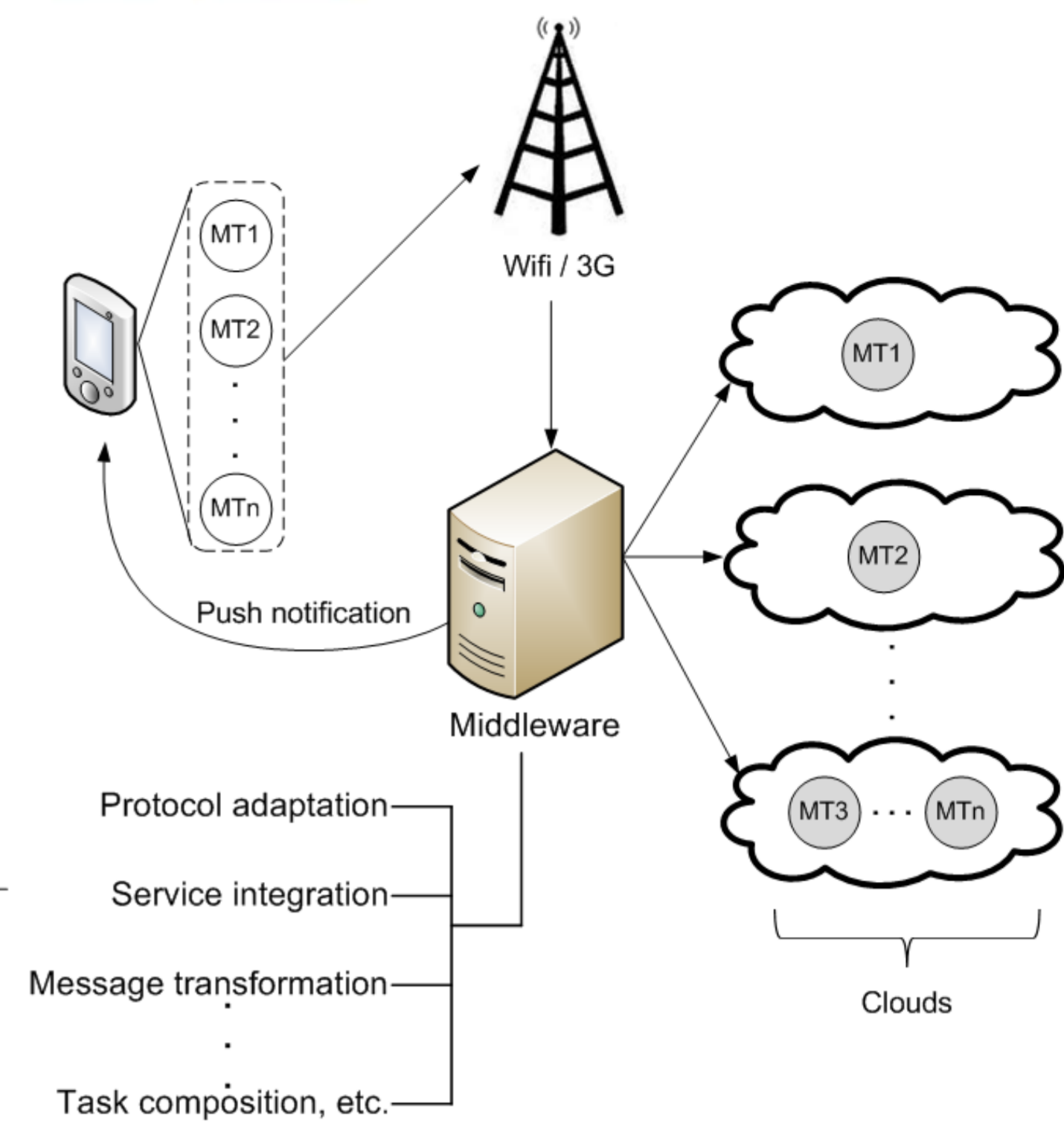
Mobile cloud computing (MCC)

Extended battery life
Increased performance
Augmented functionality

Binding mobile to cloud resources

Code offloading

Task delegation



Mobile Cloud Middleware

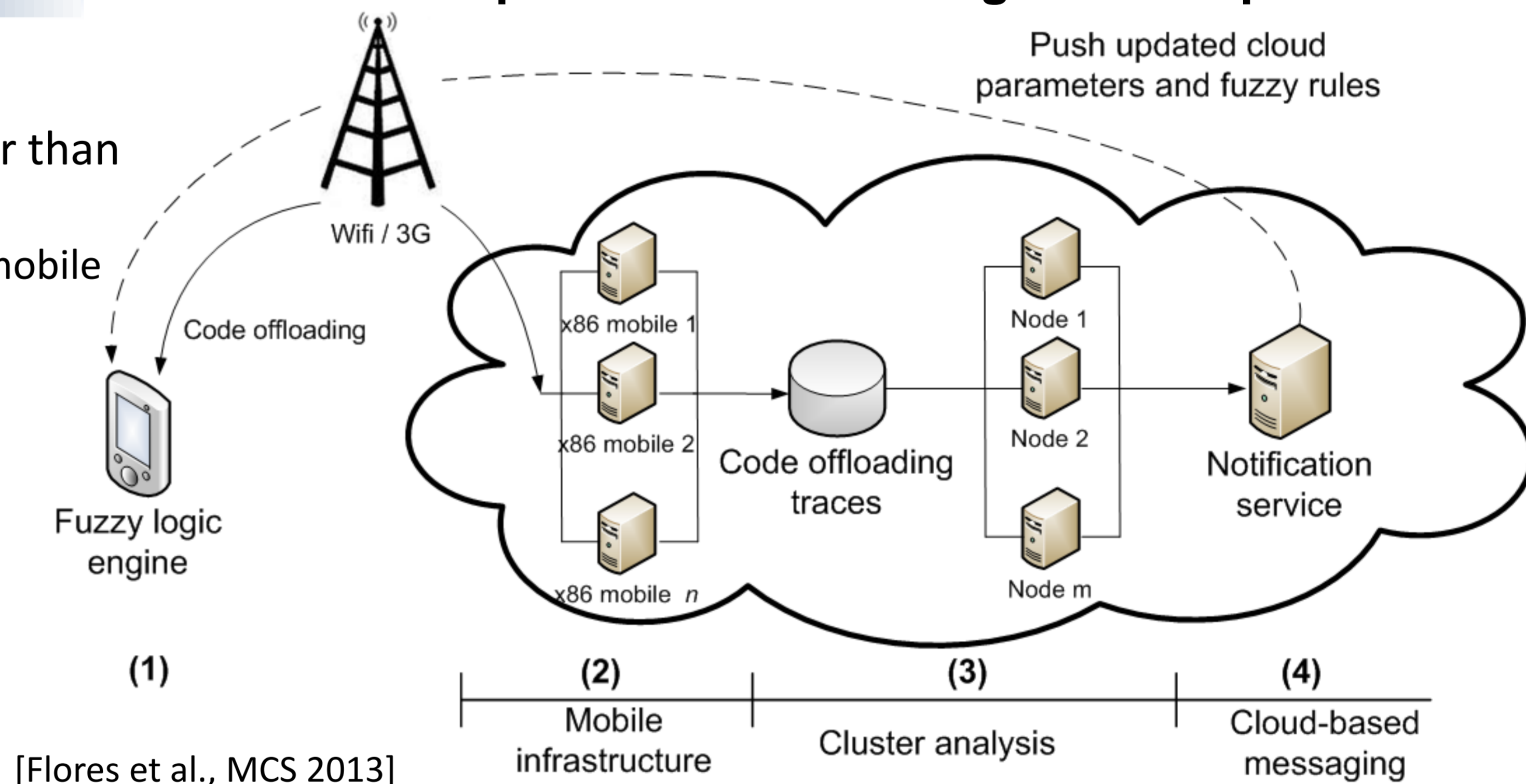
- Initial results for task delegation
 - Service integration and cloud interoperability
 - Dynamic allocation of cloud resources
 - Handling resource-intensive processing from mobiles
- [Flores et al., MoMM 2011]

Hypothesis

- Code offloading may fail?
 - Mobile component execution is non-deterministic. Thus, runtime analysis should be encouraged
 - Cloud infrastructure plays a major role
- Is mobile cloud taking full advantage of cloud computing?
- How to optimize the offloading decision process?

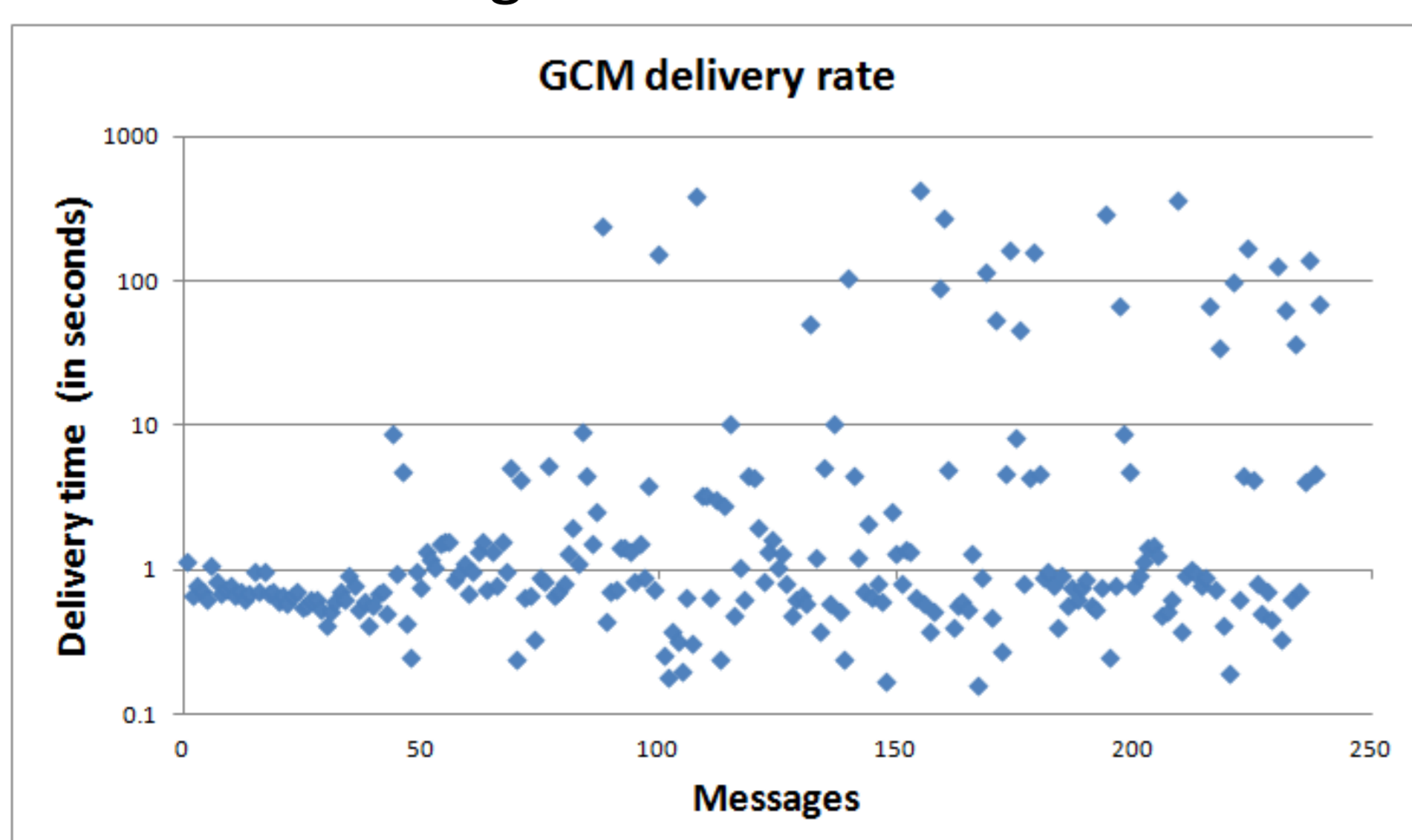
Proposed solution

- Offloading from a different perspective
 - "Offloading is a global learning process rather than local decision process"
 - Fuzzy logic engine running within the mobile
 - Collecting code traces in cloud storage
 - Analysis of code offloading traces
 - Push knowledge to the mobile
- EMCO: Evidence Based Mobile Code Offloading
 - Cloud is expert, handset asks for expertise
 - Considers mobile and cloud parameters in the offloading decision process



Results

- Use cases
 - Mobile component scheduling
 - Enhanced offloading decision process
 - Richer application partitioning, etc.
- Performance
 - Responsiveness of cloud-based push
 - technologies



Experiments

- Global inference vs. Local decision

Parallelization of mobile components?

Beneficial from a global perspective rather than a local context

