Mobile Cloud Computing

Mobile Cloud Computing

- Motivation
  - Growth in the use of Smart phones, apps
  - Increased capabilities of mobile devices
  - Access of internet using Mobile devices than PCs!
- Resource challenges (battery life, storage, bandwidth etc.) in mobile devices??
- Cloud computing offers advantages to users by allowing them to use infrastructure, platforms and software by cloud providers at low cost and elastically in an on-demand fashion
- “Information at your fingertips anywhere anytime..”

MobileBackend-as-a-service

- **What**
  - Provides mobile application developers a way to connect their application to backend cloud storage and processing
- **Why**
  - Abstract away complexities of launching and managing own infrastructure
  - Focus more on front-end development instead of backend functions
- **When**
  - Multiple Apps, Multiple Backends, Multiple Developers
  - Multiple Mobile Platforms, Multiple Integration, Multiple 3rd Party Systems & Tools
- **How**
  - Meaningful resources for app development acceleration – 3rd party API, Device SDK’s, Enterprise Connectors, Social integration, Cloud storage

Augmenting Mobiles with Cloud Computing

- Amazon Silk browser
  - Split browser
- Apple Siri
  - Speech recognition in cloud
- Apple iCloud
  - Unlimited storage and sync capabilities
- Image recognition apps on smart-phones useful in developing augmentedreality apps on mobile devices
  - Augmented reality app using Google Glass

What is Mobile Cloud Computing?

- Mobile cloud computing (MCC) is the combination of cloud computing, mobile computing and wireless networks to bring rich computational resources to mobile users.
- MCC provides mobile users with data storage and processing services in clouds
  - Obviating the need to have a powerful device configuration (e.g. CPU speed, memory capacity etc.)
  - All resource-intensive computing can be performed in the of cloud
- Moving computing power and data away from

  the mobile devices
  - Powerful and centralized computing platforms located in clouds
  - Accessed over the wireless connection based on a thin native client

Why Mobile Cloud Computing?

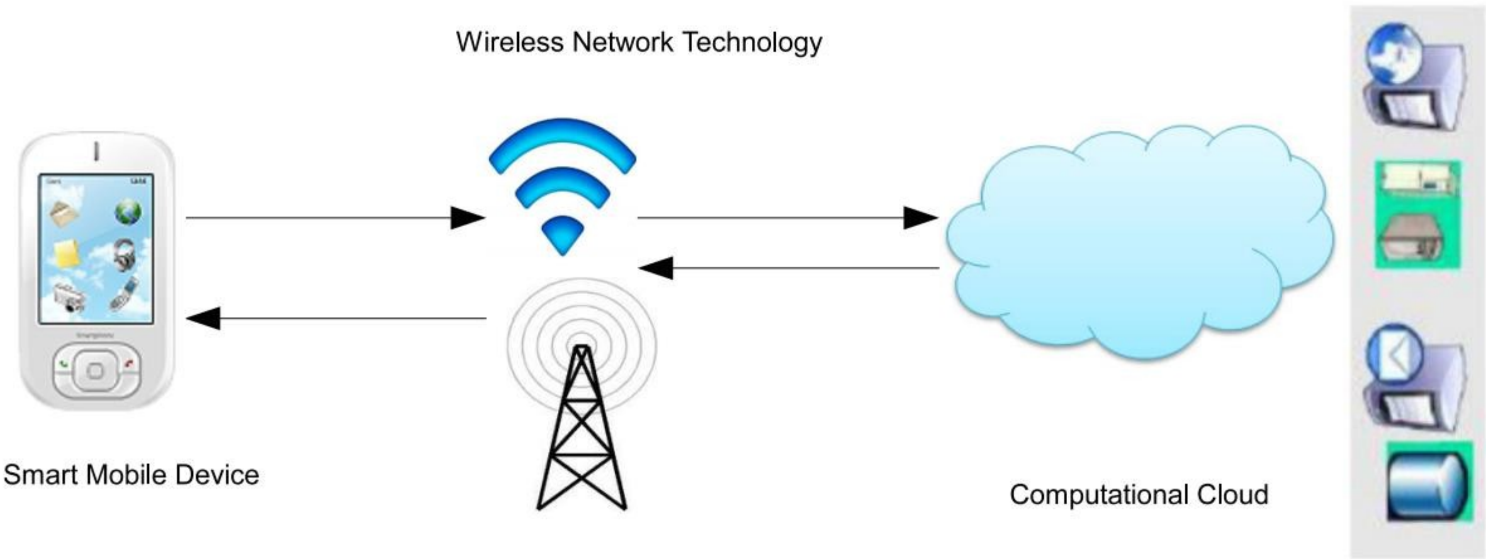
- Speed and flexibility
  - Mobile cloud applications can be built or revised quickly using cloud services. They can be delivered to many different devices with different operating systems
- Shared resources
  - Mobile apps that run on the cloud are not constrained by a device's storage and processing resources. Data-intensive processes can run in the cloud. User engagement can continue seamlessly from one device to another.
- Integrated data
  - Mobile cloud computing enables users to quickly and securely collect and integrate data from various sources, regardless of where it resides.

Key-features of Mobile Cloud Computing

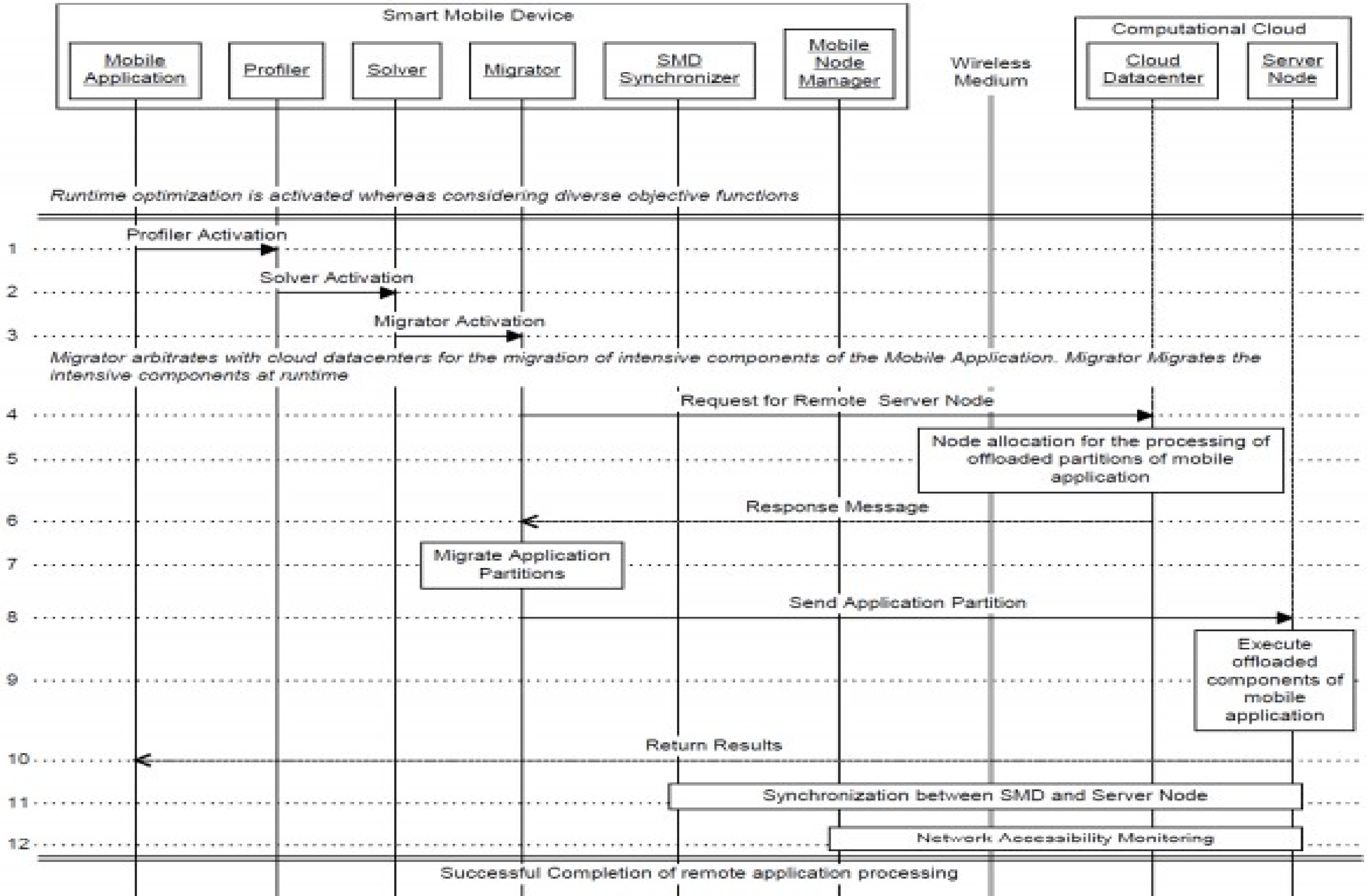
- Mobile cloud computing delivers applications to mobile devices quickly and securely, with capabilities beyond those of local resources
  - Facilitates the quick development, delivery and management of mobile apps
  - Uses fewer device resources because applications are cloud-supported
  - Supports a variety of development approaches and devices
  - Mobile devices connect to services delivered through an API architecture
  - Improves reliability with information backed up and stored in the cloud

Mobile Cloud Computing

- Mobile Cloud Computing is a framework to augment a resource constrained mobile device to execute parts of the program on cloud based servers
- Pros
  - Saves battery power
  - Makes execution faster
- Cons
  - Must send the program states (data) to the cloud server, hence consumes battery
  - Network latency can lead to execution delay



Typical MCC Workflow



Dynamic Runtime Offloading

- Dynamic runtime offloading involves the issues of
  - dynamic application profiling and solver on SMD
  - runtime application partitioning
  - migration of intensive components
  - continuous synchronization for the entire duration of runtime execution platform.

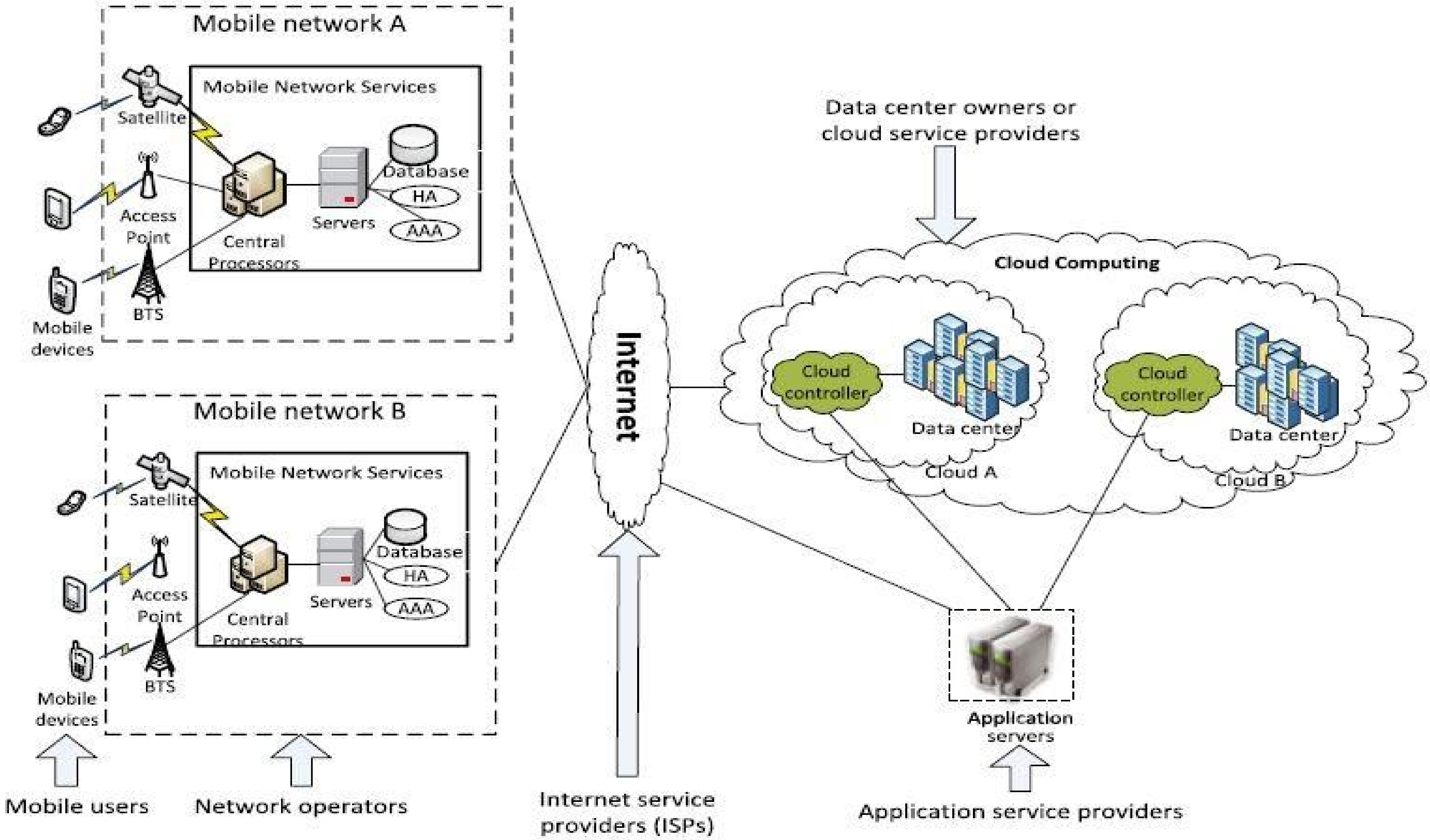
MCC key components

- Profiler
  - Profiler monitors application execution to collect data about the time to execute, power consumption, network traffic
- Solver
  - Solver has the task of selecting which parts of an app runs on mobile and cloud
- Synchronizer
  - Task of synchronizer modules is to collect results of split execution and combine, and make the execution details transparent to the user

Key Requirements for MCC

- **Simple APIs** offering access to mobile services, and requiring no specific knowledge of underlying network technologies
- **Web Interface**
- **Internet access** to remotely stored applications in the cloud

Mobile Cloud Computing – Typical Architecture



Advantages of MCC

- **Extending battery lifetime**
  - Computation offloading migrates large computations and complex processing from resource- limited devices (i.e., mobile devices) to resourceful machines (i.e., servers in clouds).
  - Remote application execution can save energy significantly.
  - Many mobile applications take advantages from task migration and remote processing
- **Improving data storage capacity and processing power**
  - MCC enables mobile users to store/access large data on the cloud.
  - MCC helps reduce the running cost for computation intensive applications.
  - Mobile applications are not constrained by storage capacity on the devices because their data now is stored on the cloud.

Advantages of MCC (contd...)

- **Improving Reliability and Availability**
  - Keeping data and application in the clouds reduces the chance of lost on the mobile devices.
  - MCC can be designed as a comprehensive data security model for both service providers and users:
    - Protect copyrighted digital contents in clouds.
    - Provide security services such as virus scanning, malicious code detection, authentication for mobile users.
  - With data and services in the clouds, then are always(almost) available even when the users are moving

Advantages of MCC

- Dynamic provisioning
- Scalability
- Multi-tenancy
  - Service providers can share the resources and costs to support a variety of applications and large no. of users.
- Ease of Integration
  - Multiple services from different providers can be integrated easily through the cloud and the Internet to meet the users’ demands .

Mobile Cloud Computing – Challenges

- MCC Security Issues
  - Protecting user privacy and data/application secrecy from adversaries is key to establish and maintain consumers’ trust in the mobile platform, especially in MCC.
  - MCC security issues have two main categories:
    - Security for mobile users
    - Securing data on clouds

Mobile Cloud Computing – Challenges

- **Security and Privacy for Mobile Users**
  - Mobile devices are exposed to numerous security threats like malicious codes and their vulnerability.
  - GPS can cause privacy issues for subscribers.
  - Security for mobile applications:
    - Installing and running security software are the simplest ways to detect security threats.
    - Mobile devices are resource constrained, protecting them from the threats is more difficult than that for resourceful devices.
  - Location based services (LBS) faces a privacy issue on mobile users' provide private information such as their current location.
  - Problem becomes even worse if an adversary knows user's important information.

Mobile Cloud Computing – Challenges

- **Security for Mobile Users**
  - Approaches to move the threat detection capabilities to clouds.
  - Host agent runs on mobile devices to inspect the file activity on a system. If an identified file is not available in a cache of previous analyzed files, this file will be sent to the in cloud network service for verification.
  - Attack detection for a smartphone is performed on a remote server in the cloud.
  - The smartphone records only a minimal execution trace, and transmits it to the security server in the cloud.

Mobile Cloud Computing – Challenges

- **Context-aware Mobile Cloud Services**
  - It is important to fulfill mobile users' satisfaction by monitoring their preferences and providing appropriate services to each of the users.
  - Context-aware mobile cloud services try to utilize the local contexts (e.g., data types, network status, device environments, and user preferences) to improve the quality of service (QoS).

Mobile Cloud Computing – Challenges

- **Network Access Management:**
  - An efficient network access management not only improves link performance but also optimizes bandwidth usage
- **Quality of Service:**
  - How to ensure QoS is still a big issue, especially on network delay.
  - CloneCloud and Cloudlets are expected to reduce the network delay.
  - The idea is to clone the entire set of data and applications from the smartphone onto the cloud and to selectively execute some operations on the clones, reintegrating the results back into the smartphone
- **Pricing:**
  - MCC involves both mobile service provider (MSP) and cloud service provider (CSP) with different services management, customers management, methods of payment and prices.
  - Business model including pricing and revenue sharing has to be carefully developed for MCC.

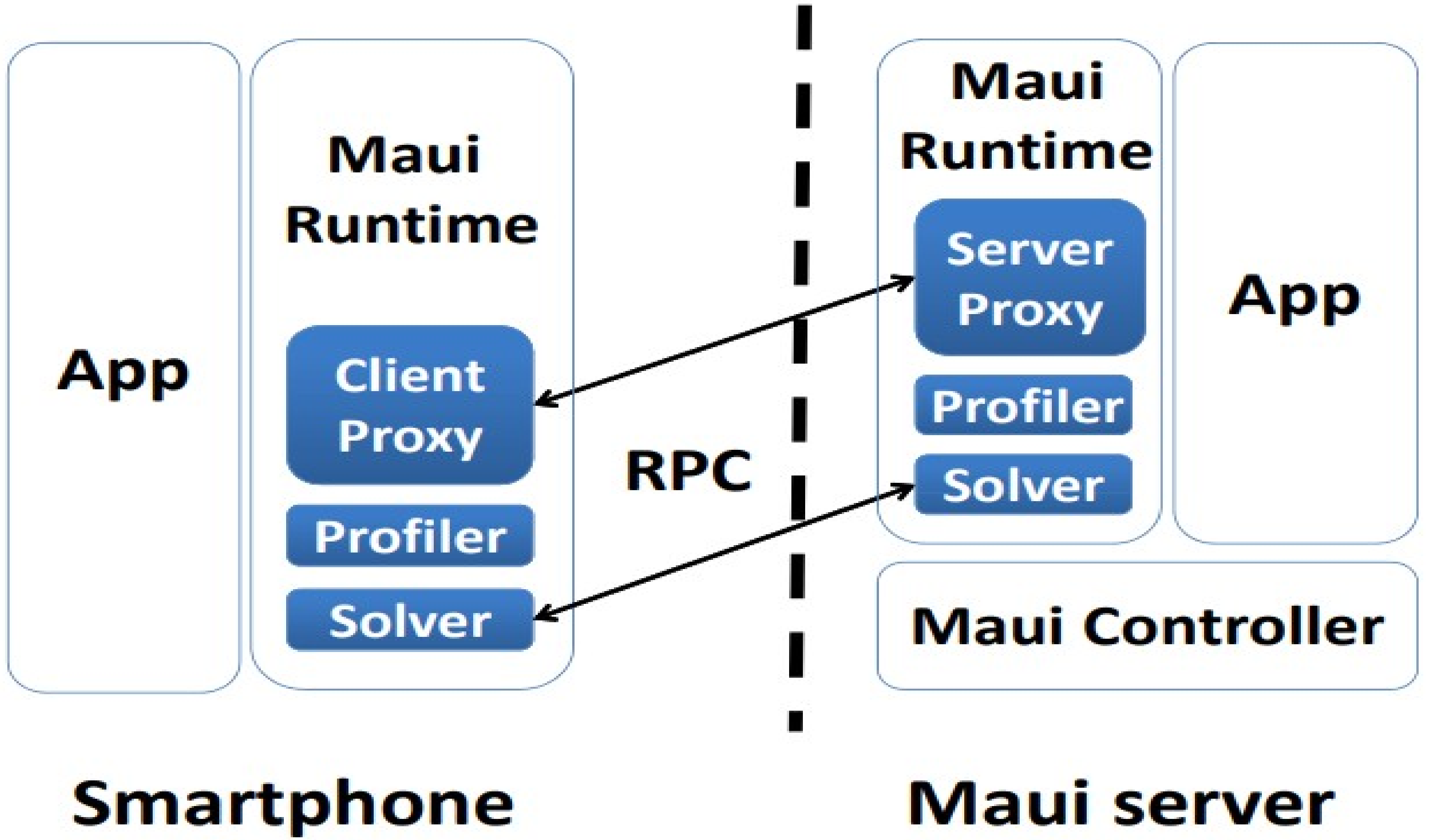
Mobile Cloud Computing – Challenges

- **Standard Interface:**
  - Interoperability becomes an important issue when mobile users need to interact with the cloud.
  - Compatibility among devices for web interface could be an issue.
  - Standard protocol, signaling, and interface between mobile users and cloud would be required.
- **Service Convergence:**
  - Services will be differentiated according to the types, cost, availability and quality.
  - New scheme is needed in which the mobile users can utilize multiple cloud in a unified fashion.
  - Automatic discover and compose services for user.
  - Sky computing is a model where resources from multiple clouds providers are leveraged to create a large scale distributed infrastructure.
  - Service integration (i.e., convergence) would need to be explored.

Key challenges

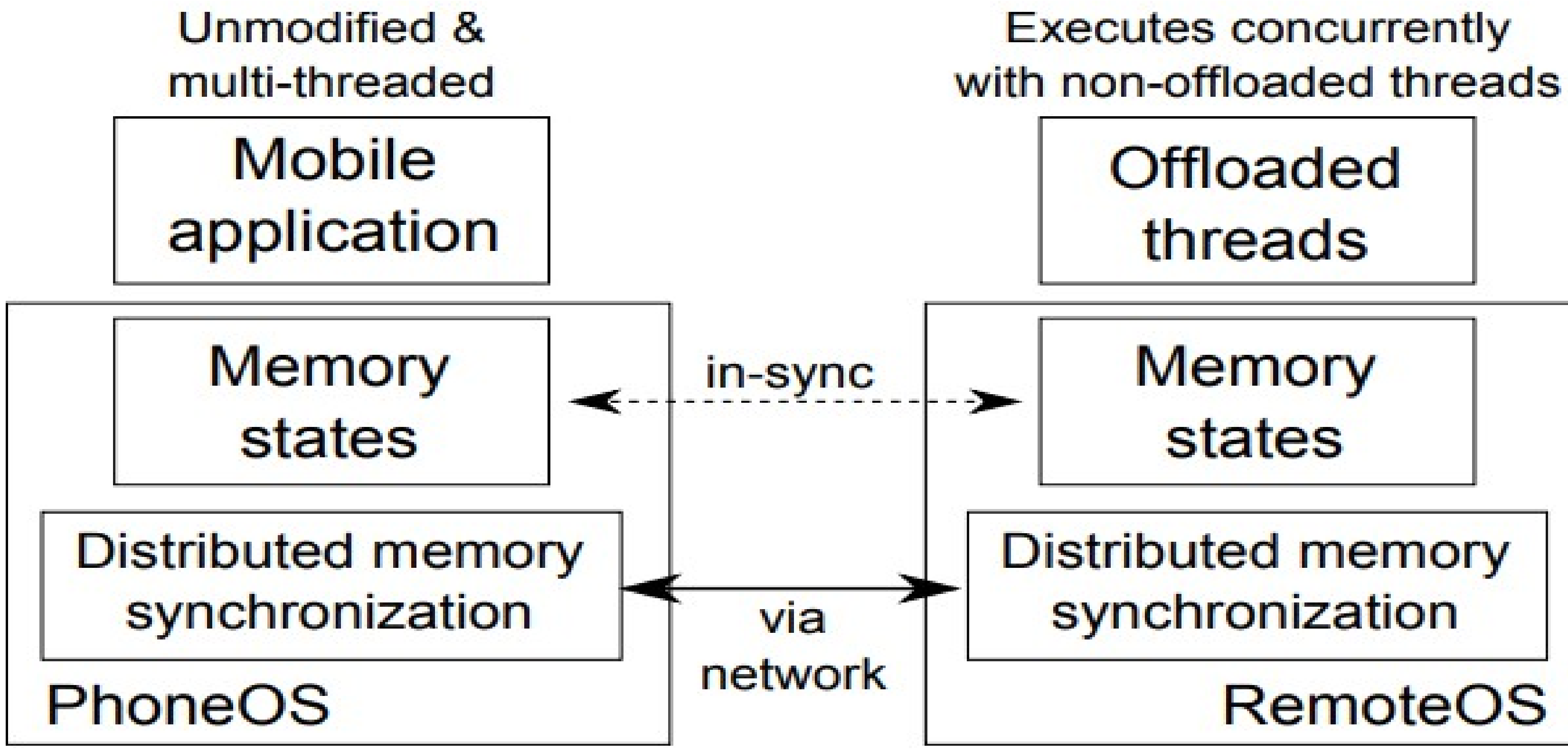
- MCC requires dynamic partitioning of an application to optimize
  - Energy saving
  - Execution time
- Requires a software (middleware) that decides at app launch which parts of the application must execute on the mobile device, and which parts must execute on cloud
  - A classic optimization problem

MCC Systems: MAUI (Mobile Assistance Using Infrastructure)



MCC Systems: COMET

- **COMET: Code Offload by Migrating Execution Transparently**



Key Problems to Solve

- At its core, MCC framework must solve how to partition a program for execution on heterogeneous computing resources
- This is a classic “Task Partitioning Problem”
- Widely studied in processor resource scheduling as “job scheduling problem”

Task Partitioning Problem in MCC

- Input:
  - A call graph representing an application's method call sequence
  - Attributes for each node in the graph denotes

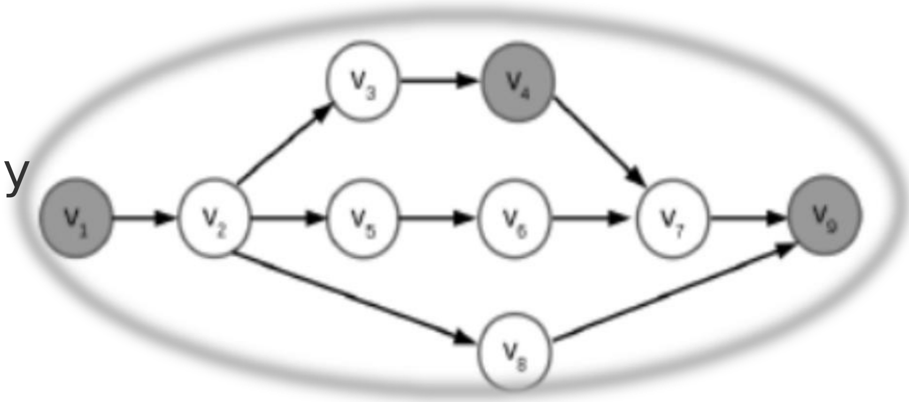
    (a) energy consumed to execute the method on the mobile device

    (b) energy consumed to transfer the program states to a remote server
- Output:
  - Partition the methods into two sets – one set marks the methods to execute on the mobile device, and the second set marks the methods to execute on cloud
- Goals and Constraints:

  1. Energy consumed must be minimized
  2. There is a limit on the execution time of the application
  3. Other constraints could be – some methods must be executed on mobile device, total monetary cost, etc.

Mathematical Formulation

- Highlighted nodes must be executed on the mobile device called native tasks (v1, v4, v9)
- Edges represent the sequence of execution – Any non-highlighted node can be executed either locally on the mobile device or on cloud
- 0-1 integer linear program,
- where I_v = 0 if method executed locally,
  - = 1 if method executed rem
- E : Energy cost to execute meth locally
- C u,v : Cost of data transfer
- L : Total execution latency
- T : Time to execute the method
- B : Time to transfer program state



$$\text{maximize} \sum_{v \in V} I_v \times E_v^l - \sum_{(u,v) \in E} |I_u - I_v| \times C_{u,v}$$

$$\text{such that:} \sum_{v \in V} ((1 - I_v) \times T_v^l) + (I_v \times T_v^r))$$

$$+ \sum_{(u,v) \in E} (|I_u - I_v| \times B_{u,v}) \leq L$$

$$\text{and} \qquad I_v \leq r_v, \ \forall v \in V$$

Static and Dynamic Partitioning

- Static Partitioning
  - When an application is launched, invoke an ILP solver which will tell where each method should be executed
  - There are also heuristics to find solutions faster
- Dynamic or Adaptive Partitioning
  - For a long running program, the environmental conditions can vary
  - Depending on the input, the energy consumption of a method can vary

Mobile Cloud Computing – Challenges/ Issues

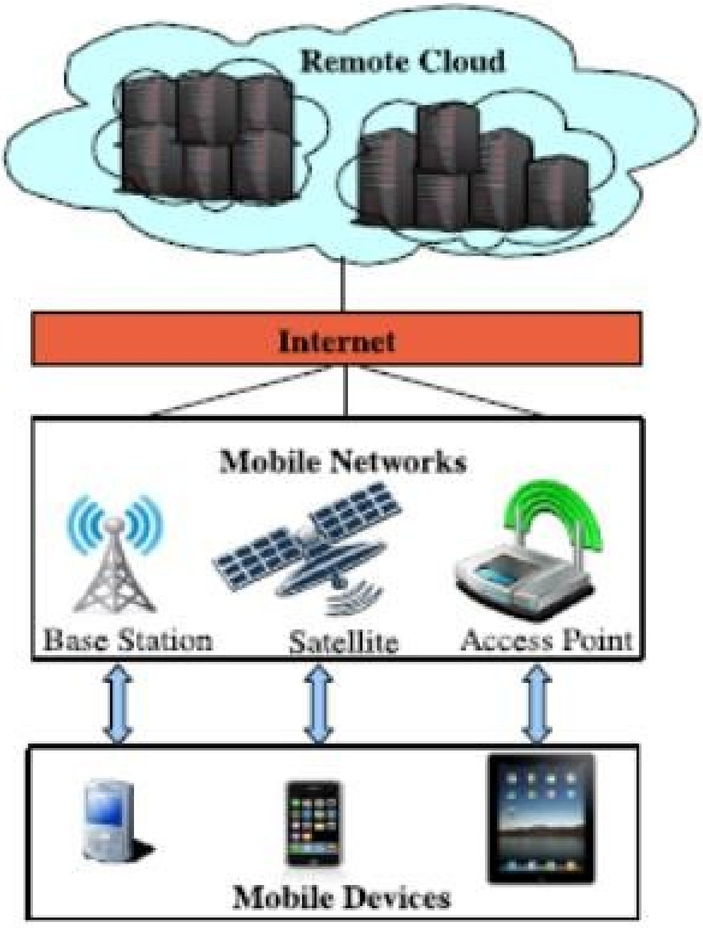
- **Mobile communication issues**
  - **Low bandwidth:** One of the biggest issues, because the radio resource for wireless networks is much more scarce than wired networks
  - **Service availability:** Mobile users may not be able to connect to the cloud to obtain a service due to traffic congestion, network failures, mobile signal strength problems
  - **Heterogeneity**: Handling wireless connectivity with highly heterogeneous networks to satisfy MCC requirements (always-on connectivity, on-demand scalability, energy efficiency) is a difficult problem
- **Computing issues (Computation offloading)**
  - One of the main features of MCC
  - Offloading is not always effective in saving energy
  - It is critical to determine whether to offload and which portions of the service codes to offload

CODE OFFLOADING USING CLOUDLET

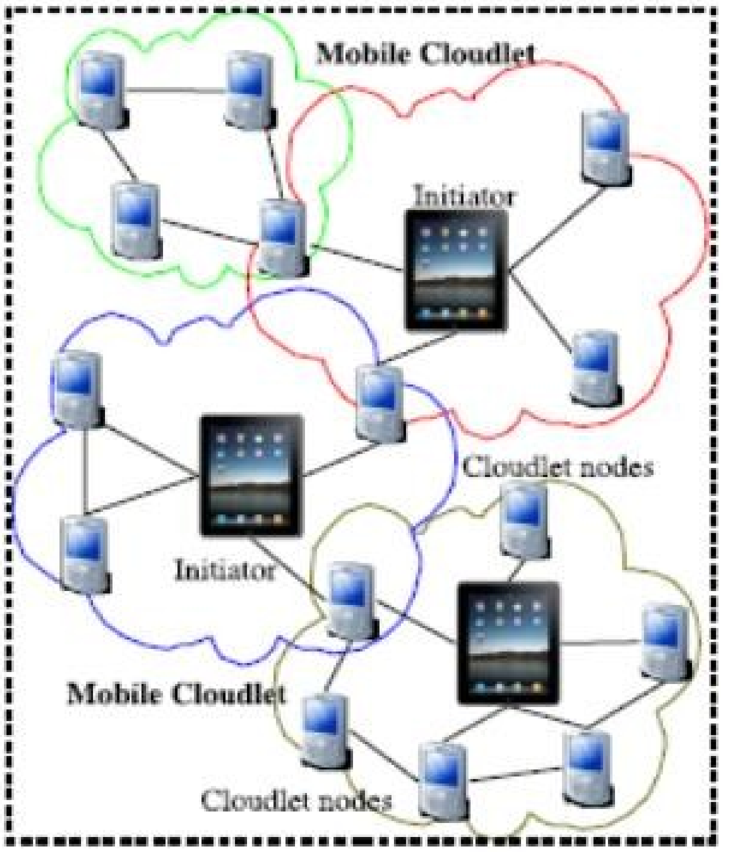
- **CLOUDLET:**
  - “a trusted, resource-rich computer or cluster of computers that is well-connected to the Internet and is available for use by nearby mobile devices.”
- **Code Offloading :**
  - Offloading the code to the remote server and executing it.
  - This architecture decreases latency by using a single-hop network and potentially lowers battery consumption by using Wi-Fi or short-range radio instead of broadband wireless which typically consumes more energy.

CODE OFFLOADING USING CLOUDLET

- **Cloudlet**



Use remote cloud



Use cloudlet

When to Offload ?

- S: Speed of cloud to compute C instructions
- M: Speed of mobile to compute C instructions
- D: Data need to transmit
- B: Bandwidth of the wireless Internet
- P_c : Energy cost per second when the mobile phone is doing computing
- P_i: Energy cost per second when the mobile phone is idle.
- P_{tr} : Energy cost per second when the mobile is transmitting the data.

Amount of energy saved is :

$$P_c \times \frac{C}{M} - P_i \times \frac{C}{S} - P_{tr} \times \frac{D}{B}$$

Suppose the server is F times faster—S= F × M.

We can rewrite the formula as

$$\frac{C}{M} \times \left(Pc - \frac{P_i}{F}\right) - P_{tr} \times \frac{D}{B}$$

When to Offload ? (contd..)

- Energy is saved when the formula produces a positive number. The formula is positive if D/B is sufficiently small compared with C/M and F is sufficiently large.
- Cloud computing can potentially save energy for mobile users.
- Not all applications are energy efficient when migrated to the cloud.
- Cloud computing services would be significantly different from cloud services for desktops because they must offer energy savings.
- The services should consider the energy overhead for privacy, security, reliability, and data communication before offloading.

The amount of energy saved is :

$$P_c \times \frac{C}{M} - P_i \times \frac{C}{S} - P_{tr} \times \frac{D}{B}$$

We can rewrite the formula as

$$\frac{C}{M} \times \left(Pc - \frac{P_i}{F}\right) - P_{tr} \times \frac{D}{B}$$

When to Offload ? (contd..)

- Energy is saved when the formula produces a positive number. The formula is positive if D/B is sufficiently small compared with C/M and F is sufficiently large.
- Cloud computing can potentially save energy for mobile users.
- Not all applications are energy efficient when migrated to the cloud.
- Cloud computing services would be significantly different from cloud services for desktops because they must offer energy savings.
- The services should consider the energy overhead for privacy, security, reliability, and data communication before offloading.

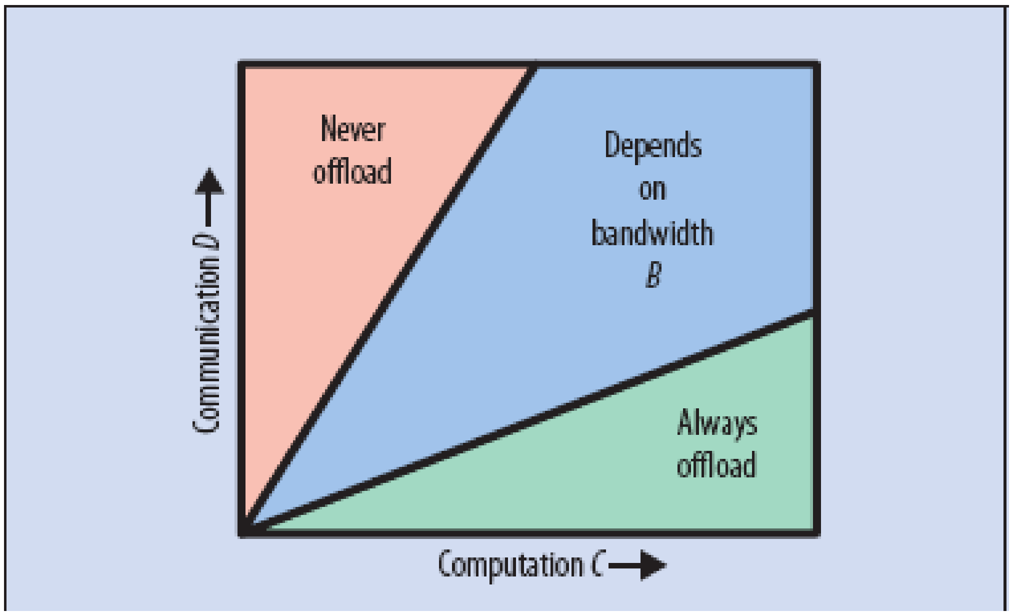
The amount of energy saved is :

$$P_c \times \frac{C}{M} - P_i \times \frac{C}{S} - P_{tr} \times \frac{D}{B}$$

We can rewrite the formula as

$$\frac{C}{M} \times \left(Pc - \frac{P_i}{F}\right) - P_{tr} \times \frac{D}{B}$$

When to Offload?? (contd..)



The amount of energy saved is :

$$P_c \times \frac{C}{M} - P_i \times \frac{C}{S} - P_{tr} \times \frac{D}{B}$$

We can rewrite the formula as

$$\frac{C}{M} \times \left(Pc - \frac{P_i}{F}\right) - P_{tr} \times \frac{D}{B}$$

Offloading is beneficial when large amounts of computation C are needed with relatively small amounts of communication D

Computation Offloading Approaches

- Partition a program based on estimation of energy consumption before execution
- Optimal program partitioning for offloading is dynamically calculated based on the trade-off between the communication and computation costs at run time.
- Offloading scheme based on profiling information about computation time and data sharing at the level of procedure calls.
  - A cost graph is constructed and a branch-and-bound algorithm is applied to minimize the total energy consumption of computation and the total data communication cost.

How to evaluate MCC performance

- Energy Consumption
  - Must reduce energy usage and extend battery life
- Time to Completion
  - Should not take longer to finish the application compared to local execution
- Monetary Cost
  - Cost of network usage and server usage must be optimized
- Security
  - As offloading transfers data to the servers, ensure confidentiality and privacy of data, how to identify methods which process confidential data