

Operating System Issues in Wireless Ad-Hoc Networks

Kimmo Raatikainen

Professor, University of Helsinki

**Principal Scientist, Helsinki Institute for Information
Technology**

Research Fellow, Nokia Research Center

kimmo.raatikainen@{cs.helsinki.fi,hiit.fi,nokia.com}

<http://www.cs.helsinki.fi/Kimmo.Raatikainen/>

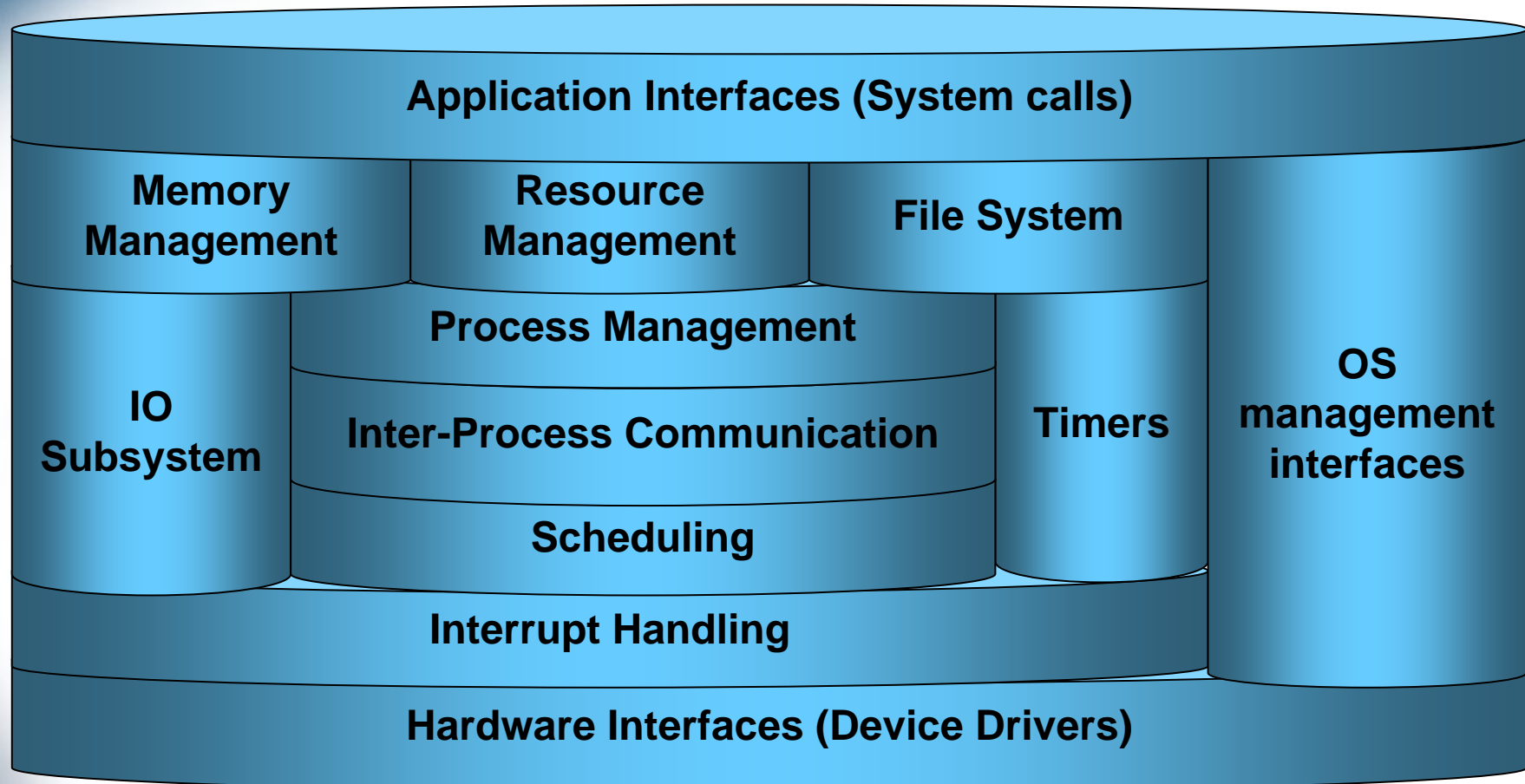
Disclaimer

**These views
Some research only reflect
Of Nokia Research Center
University of Helsinki and
not any product plans
of the Nokia Group**

Is There Still Open Research Issues in the Operating Systems?

- Operating Systems have been 50 years in the core of Computer Science
- Operating System structure and functionality is well established

Operating System Reference Model



Is There Still Open Research Issues in the Operating Systems?

- Operating Systems have been 50 years in the core of Computer Science
- Operating System structure and functionality is well established
- But I see fundamental changes in operational requirements

Underlying Assumptions

- Future applications will be
 - ***context-sensitive,***
 - ***adaptive, and***
 - ***personalized.***
- Applications need to be executed, in a reasonable and secure manner, on variety of execution environments:
 - ***anywhere,***
 - ***anyhow,***
 - ***anytime, and***
 - ***by anybody***

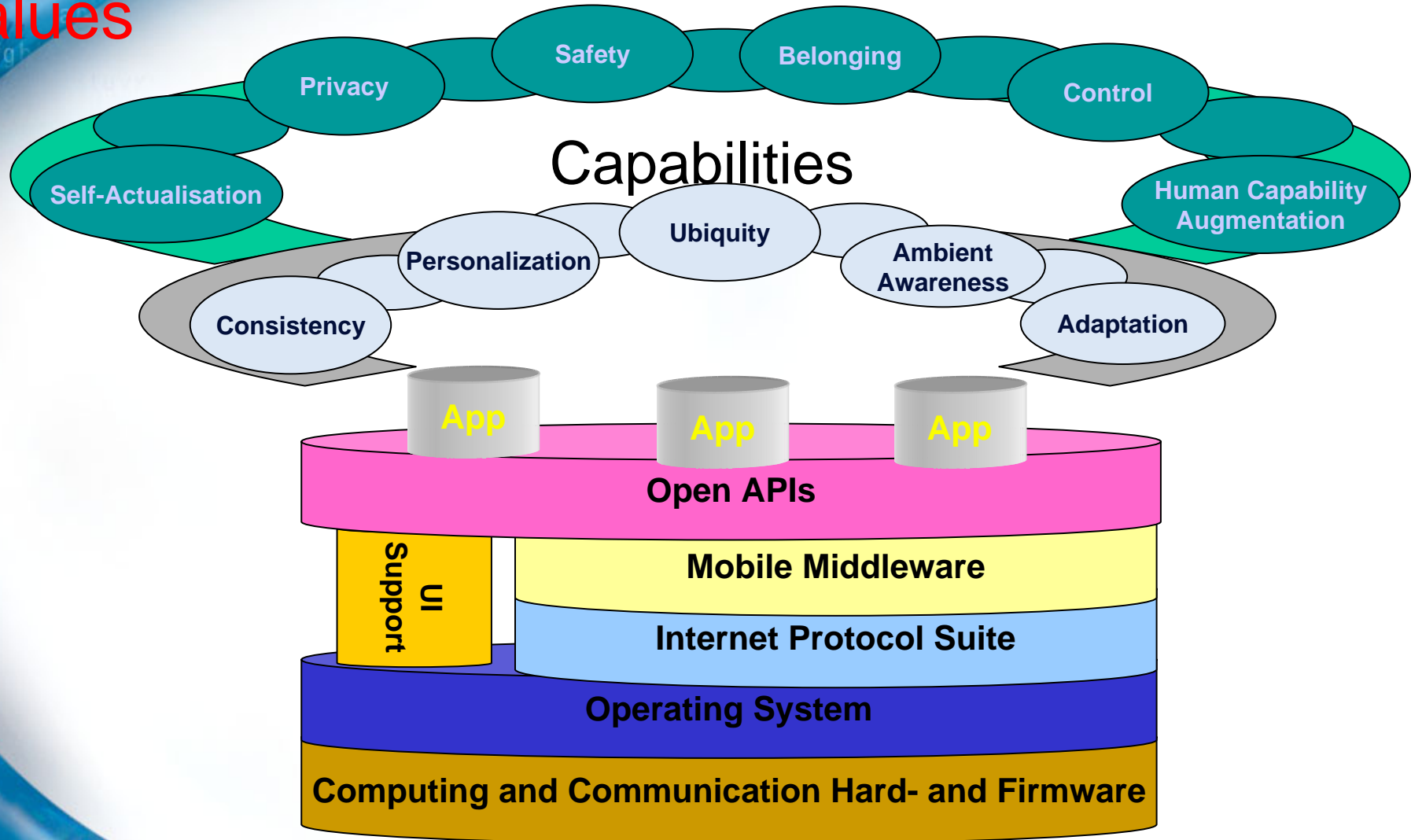
Underlying Assumptions

- Future applications will be
 - *context-sensitive*
 - *adaptive*
 - *personalized*
- Applications must be executed in a reasonable and predictable manner, on various execution environments:
 - *anywhere*
 - *anytime*, and
 - *by anybody*

What kind of operating system support is needed?

Solution Stack

Values



Next Generation Operating Systems Need to Be Designed for Reconfigurability

- Forget end-user terminals
- Start to think about end-user systems

Functionality in Reconfigurable Systems

- environment monitoring, device detection
- service discovery
- event notification and filtering
- hardware and software configuration management
 - autoconfiguration
- decisions engine
 - when and how to reconfigure
 - modeling and learning capabilities
- maintaining system integrity

Functionality in Reconfigurable Systems

- environment monitoring, fault detection
- service discovery
- error classification, handling
- hardware/software co-simulation and management
- autoconfiguration
- engine
- when to reconfigure
- learning and learning capabilities
- maintaining system integrity

Everything is not done on the operating system level but almost anything needs operating system support

Five Research Issues in Operating Systems

Research Issue 1: System Integrity

- How to verify a new piece of software?
 - Can we rely on it?
 - Is it trustworthy?
- Reliable bootstrap and re-configuration
- Controlled DMA
 - Applications run logic of their own in display processor, FPGA, ...
- Is the approach by Trusted Computing Group enough?

Research Issue 2: Power Management

- Power density cannot significantly increase
 - advances in materials and batteries will help, BUT
- We need enhanced power control
 - Monitoring available energy and its consumption
 - Predicting energy consumption per tasks
 - End-user understandable decision rules for execution:
 - what, where, when

Research Issue 3: Self-awareness

- Model of hardware and software configuration
 - to describe
 - to reason
- Reference models

Research Issue 4: Distribution

- Single Distributed Operating System
- Multiple Co-operating Operating Systems
- What about sharing
 - two end-user systems sharing a set of box, ...
- System autonomy

Research Issue 5: Detection and notifications

- Frequent interrupts
- Event filtering

- Multiple notification targets
 - multicasting, group communication,
 - implications to scheduling

Roadmap

Let Us Repeat the x-Kernel/TinyOS Exercise

- *They did a paradigm shift*
- x-kernel was designed on the terms of network communication
 - Hutchinson, N.C. and Peterson, L.L.: The x-Kernel: an architecture for implementing network protocols, IEEE Transactions on Software Engineering, 17, 1 (Jan. 1991), pp. 64-76.
- TinyOS was designed around event handling in sensor networks
 - Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler and Kristofer Pister: System Architecture Directions for Networked Sensors, ASPLOS 2000, Cambridge, MA, Nov. 12-15, 2000, pp. 93-104.

Roadmap

- Forget existing operating systems
- Design for new requirements
 - that is for reconfigurability
- Evaluate possibilities of reuse

- Replacing legacy is today expensive
 - but replacing it tomorrow is much more expensive
- The **ONLY** tricky thing is timing



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI



It is time to go back to the basics!