

An Operating System for the Home



Colin Dixon (IBM Research) Ratul Mahajan Sharad Agarwal
A.J. Brush Bongshin Lee Stefan Saroiu Paramvir Bahl

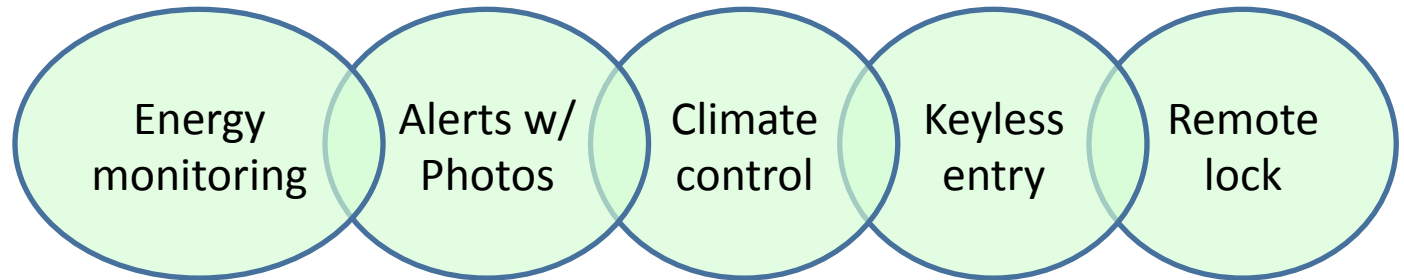
Microsoft®
Research

HomeOS

- PC-like organization for tech in the home
 - Ease management and extensibility
- Running in 12 *real* homes for 4–8 months
- Used by 42 student developers at 10 institutions

Where's my smarthome?

Tasks
(software)



Devices
(hardware)



Gap between potential and reality



Envisioned by many researchers and companies
Struggling to break into the mainstream
— Despite commercial availability since 1970s

Understanding the gap

- Study of homes with modern automation
 - 31 people across 14 households
 - Enjoyed convenience, peace of mind and control
 - But, had difficulty in two key areas:

Poor extensibility



Management pain



Existing abstractions for home tech

Network of devices

Management is still hard

- Users must manage each device/task
- Developers must deal directly w/ h/w



Appliance

Extensibility is still hard

- Closed set of tasks
- Closed set of devices

Remote
monitoring



Climate
control

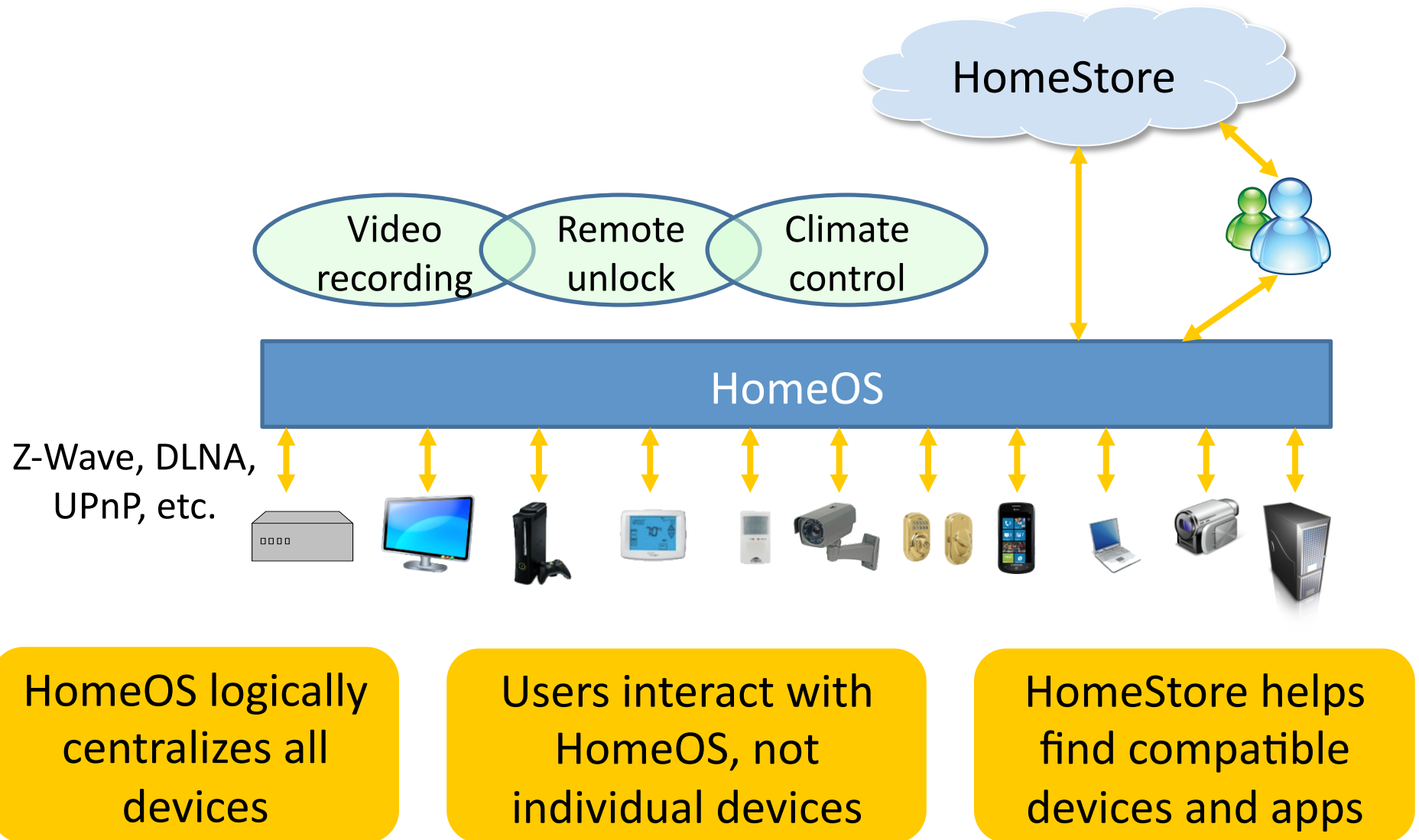


The home as a PC

View the home as a computer

- Networked devices \approx peripherals (w/drivers)
- Tasks over these devices \approx applications
- Adding devices \approx plugging in a peripheral
- Adding tasks \approx installing an application
- Managing networked devices \approx managing files

HomeOS: An OS for the home



Challenges in the home

Manageability Non-expert users must become network managers

- Need rich, but easy to use management tools
- *E.g., misconfigured app may be able to unlock a door*

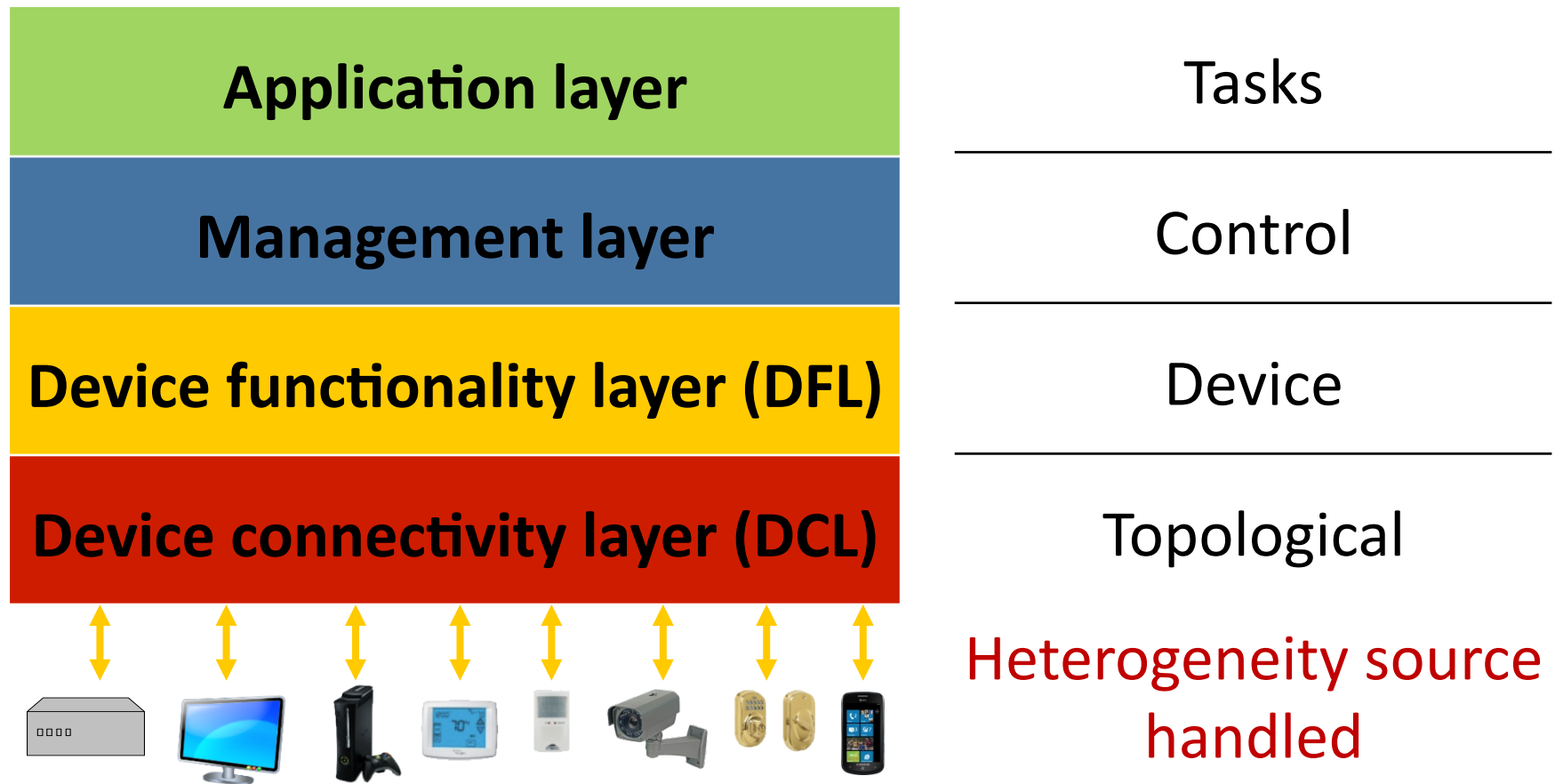
Extensibility Developers struggle to build apps

- Heterogeneity in tasks, control, device and topology

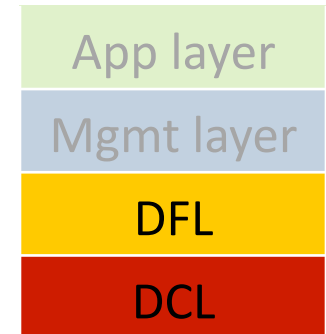
New classes of devices arrive frequently

- *E.g., Kinect, energy meters, connected TVs, etc.*

HomeOS architecture



DCL and DFL (Drivers)



DCL provides basic connectivity to devices

DFL exports device functionality as a service

- Services are protocol-independent
- Identified using roles and operations
- Kernel does not parse or understand services

Layer of Indirection between protocols and apps

Dimmer

Set(level)

Get() → level

PTZ Camera

GetImage() → bitmap

Up(), Down()

Left(), Right()

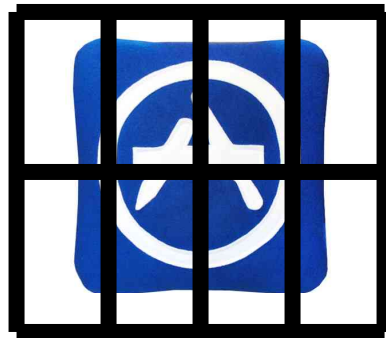
ZoomIn(), ZoomOut()

Management Layer Requirements

**Time-based
access control**



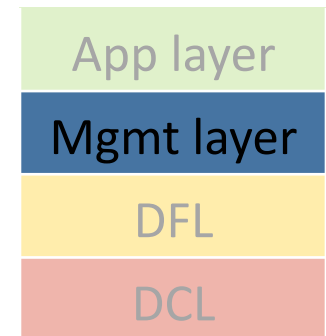
**Apps as security
principals**



**Easy-to-verify
settings**



Management Layer



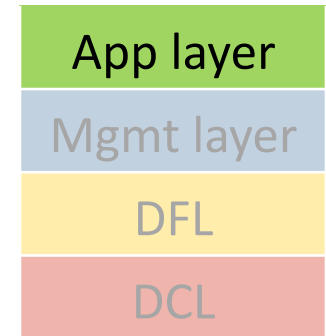
Access control policy:

- Datalog-based rules
 - *[user group, device, app, t_{start} , t_{end} , dayOfWeek, priority, accessMode]*
- Rules include ***time*** and ***applications***
- Allow users to query rules to verify their intent

Easier to reason about than ACLs in current OSes

Scales better than 2-D grid of users and devices

Application layer



Apps consume and compose abstract device interfaces from the DFL

Management layer interposes on accesses

Manifests help with compatibility testing

- Lists of mandatory and optional features
- E.g., mandatory: *{TV, SonyTV}, {MediaServer}*
optional: *{Speaker}*

Demo

Evaluating HomeOS

Key questions:

- Can non-technical users manage HomeOS?
- Can developers easily write apps and drivers?

Method:

- Field experiences
 - 12 real homes and 42 student developers
- Controlled experiments

Field experiences: The good

Users could manage their HomeOS deployments

Users particularly liked the ability to organically extend their technology

Developers found the programming abstractions and layering to be “natural”

Sample third-party applications



For more, see the HomeOS site:
<http://research.microsoft.com/homeos/>

Field experiences: The bad

Users found it hard to diagnose faults

Interoperability protocols can be fragile

Not all device features may be exposed over the network

Controlled Evaluations

10 developers asked to write one of two realistic apps

- “music follows the lights” or “custom lights per user”
- No prior experience with HomeOS
- 8 finished in under 2 hours

12 non-expert users given 7 representative mgmt. tasks

- No training with management interface
- 77% completion rate; 89% after removing an outlier task

Performance results in the paper

Conclusions

HomeOS eases extensibility and management by providing a PC abstraction for home technology

Still lots of exciting things to do!

- What core capabilities should be in every home?
- Can we provide non-intrusive identity inference?

For more info and to request code:

<http://research.microsoft.com/homeos/>