

go Bezirk

Things plus Cloud does not equal IoT



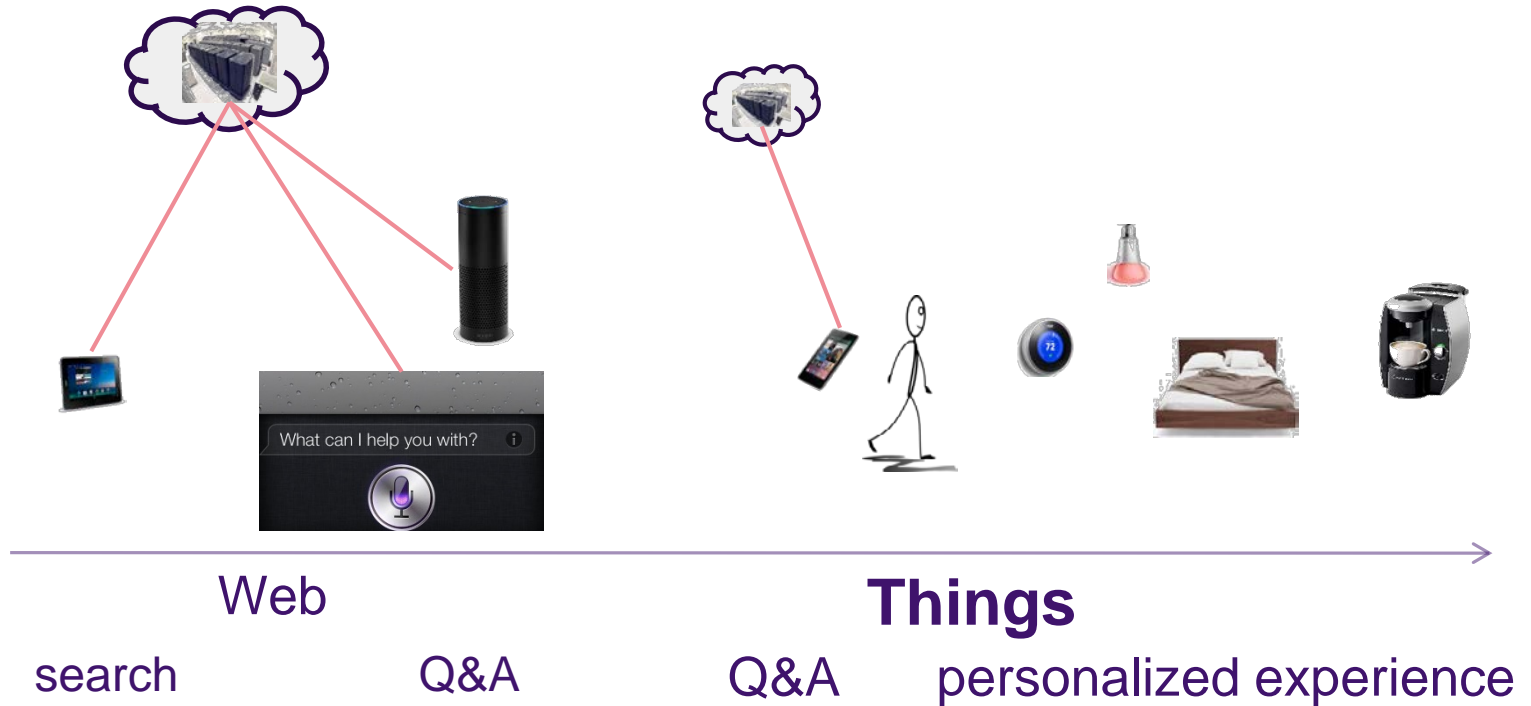
IoT by default



IoT that tastes better

problem

Architecting the IoT (experienced by people)

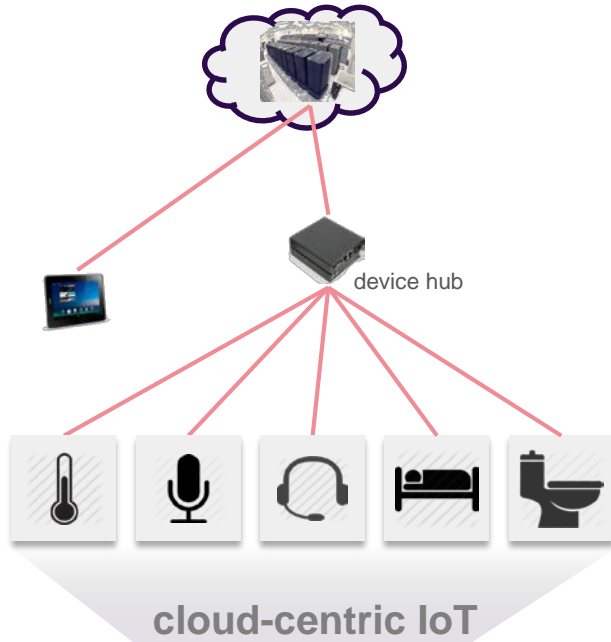


everywhere

Turn lights on
Why are they red?
Get me a coffee!

problem

Architecting the IoT



By default

→ raw data pumped to the cloud for processing and analytics

Reality check

→ responsiveness

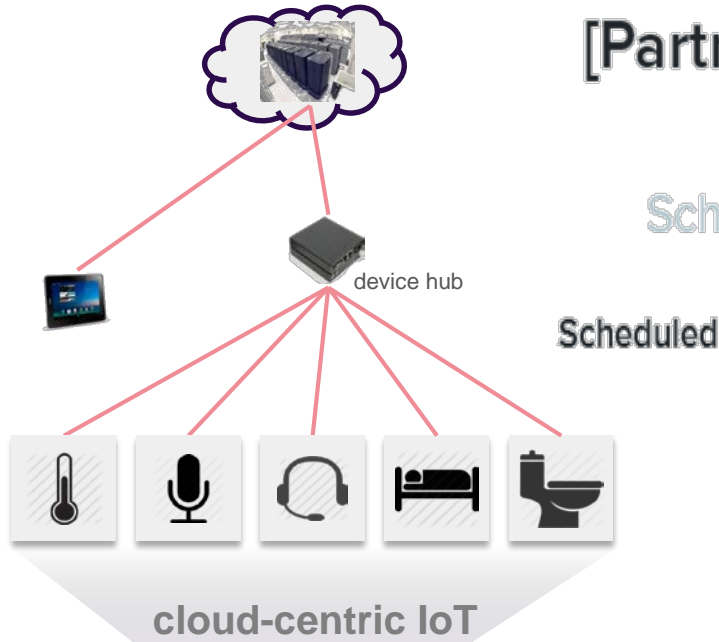
→ multi-vendor fragmentation

→ rampant threats to privacy



IoT by default responsiveness

Reality check



[Partner Maintenance] Chamberlain MyQ garage doors Scheduled Maintenance Report for Wink

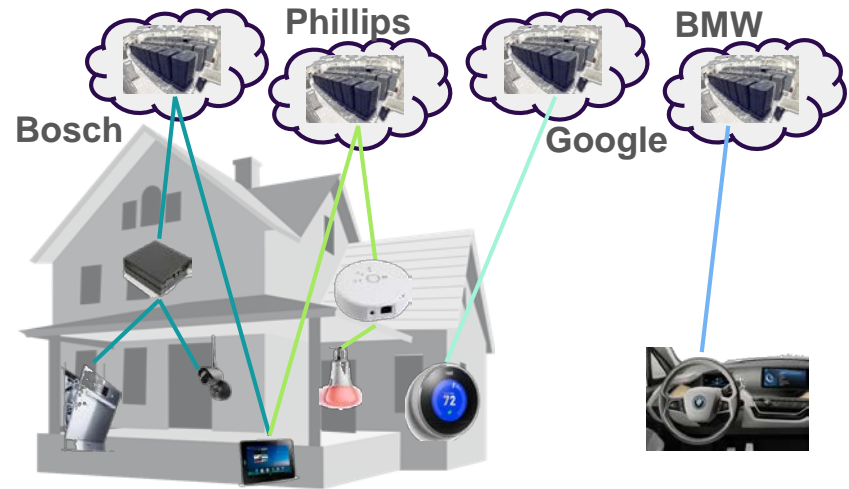
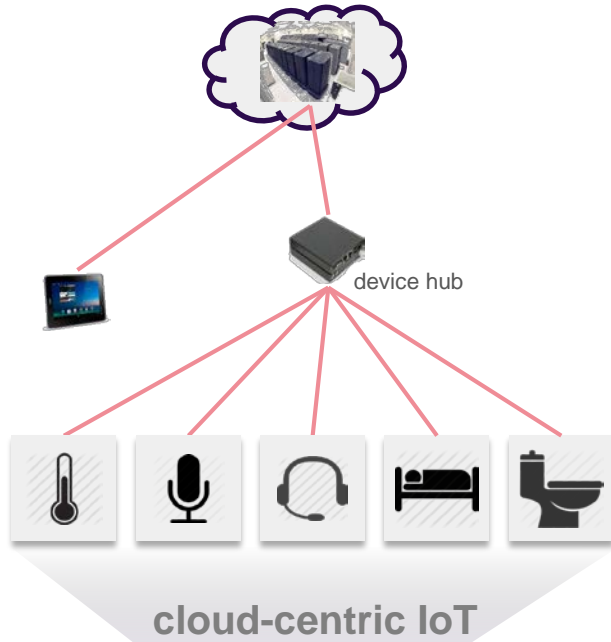
We've been informed that the Chamberlain MyQ garage door web services will be undergoing maintenance on Monday, Nov. 10 01:00 EST (Nov. 9, 22:00 PST). The maintenance is expected to last about 4 hours and users will not be able to use the Chamberlain garage doors from the app during this time.

Posted 3 months ago, Nov 08, 2014 - 22:52 UTC



IoT by default multi-vendor fragmentation

Reality check



- mainstream **business models** revolve on the **value of data** for service providers
- **Data** becomes a business asset: little incentive to share



IoT

not all **Data** is created equal

Public & corporate data:

weather, traffic,
shopping, customer support...



shared



owned



Social networks:

friends, pictures...

Owned devices:

energy usage,
maintenance diagnostics...



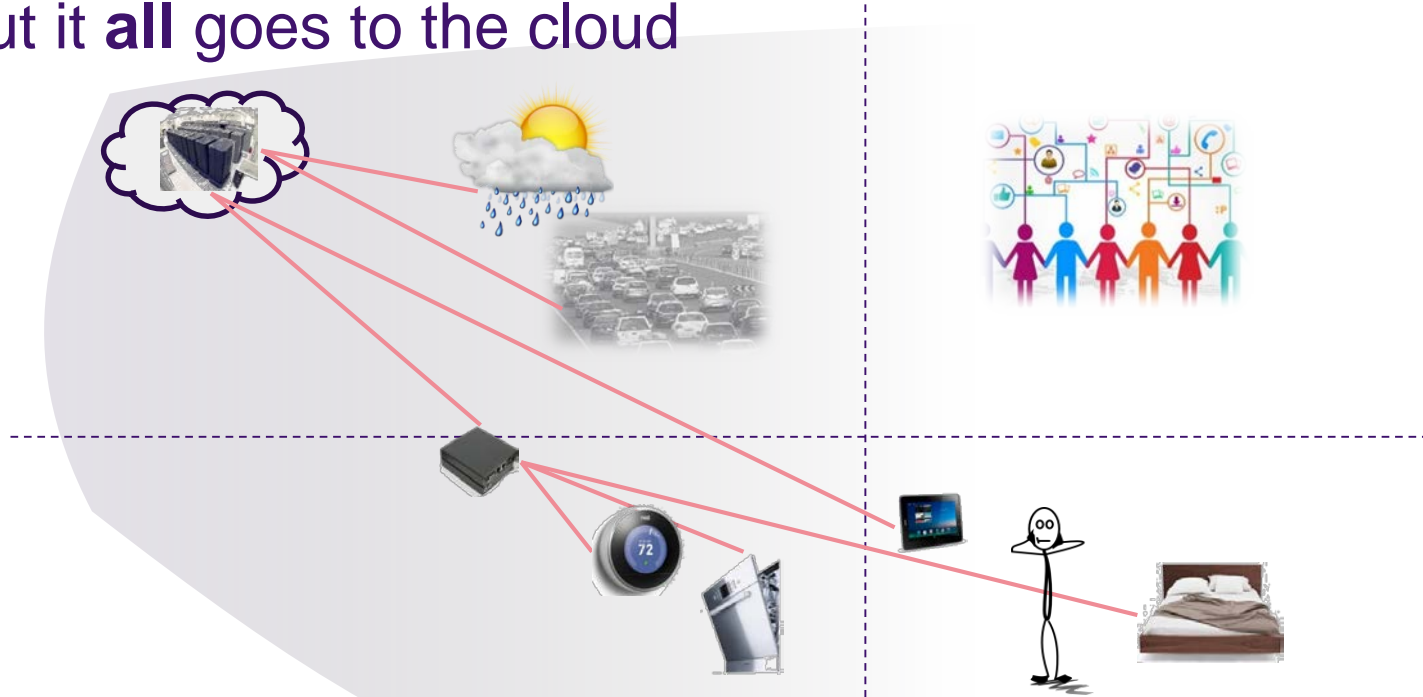
User experience:

how did you sleep?
what are you doing?
what are you asking?

things

you

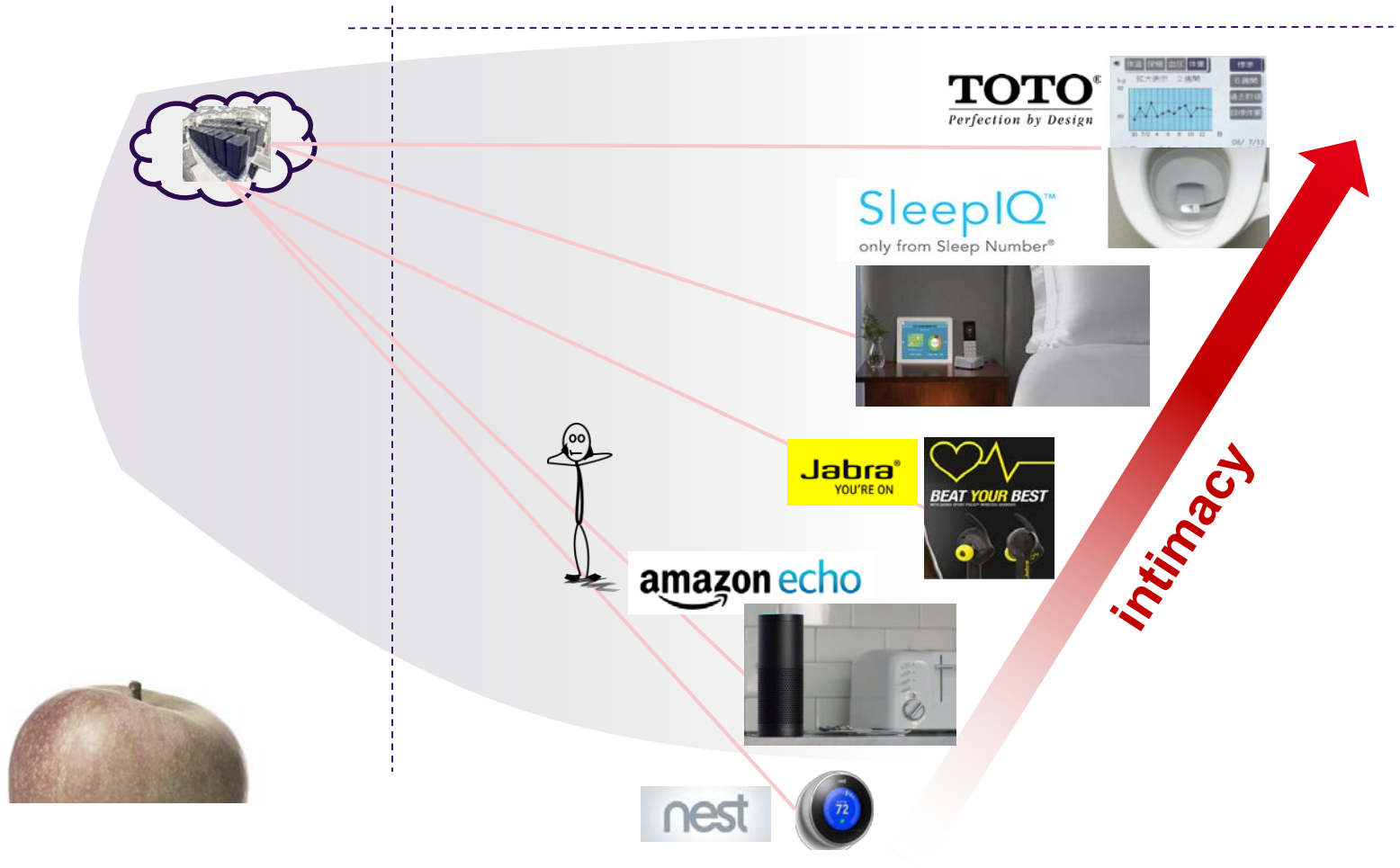
IoT
not all **Data** is created equal
but it **all** goes to the cloud



what does that mean to **you**?

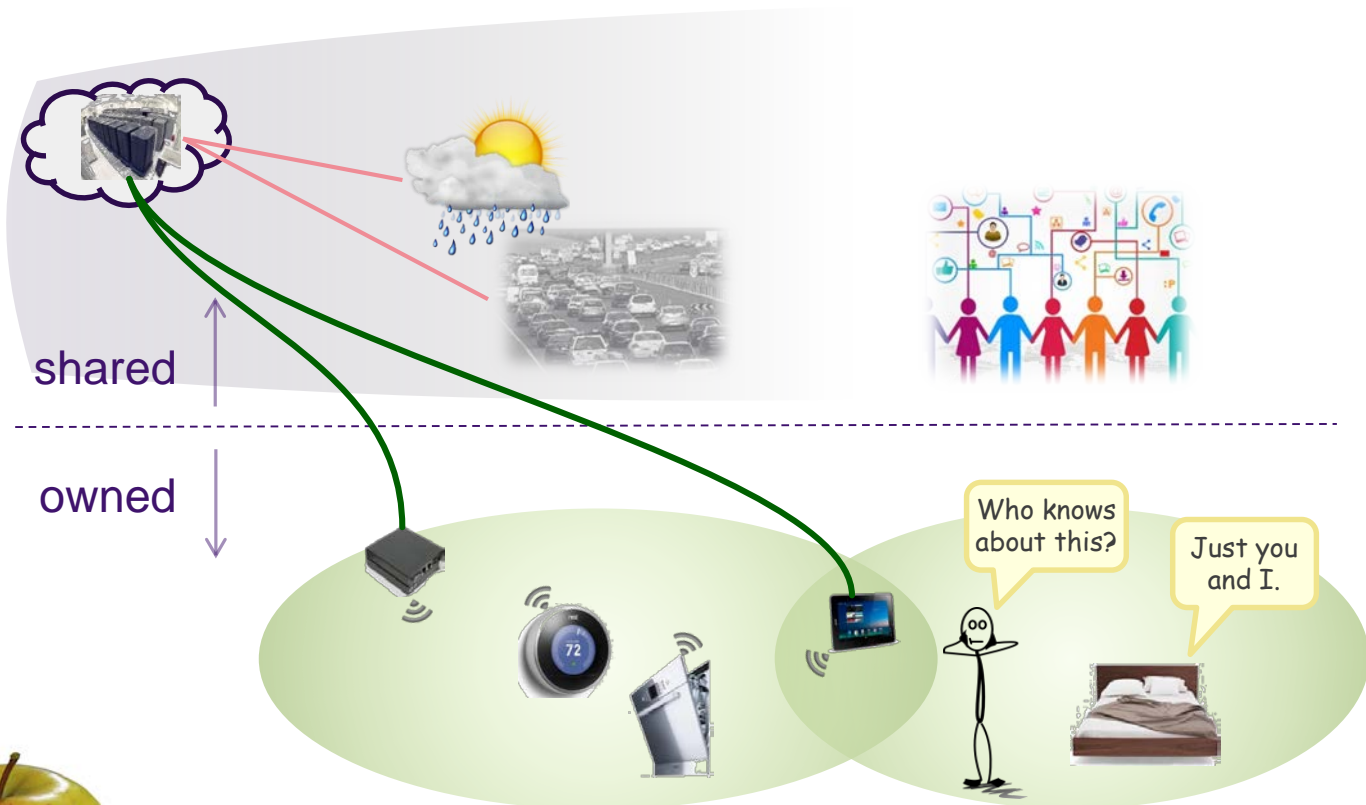
IoT by default rampant loss of **privacy**

Reality check



Bezirk *is*

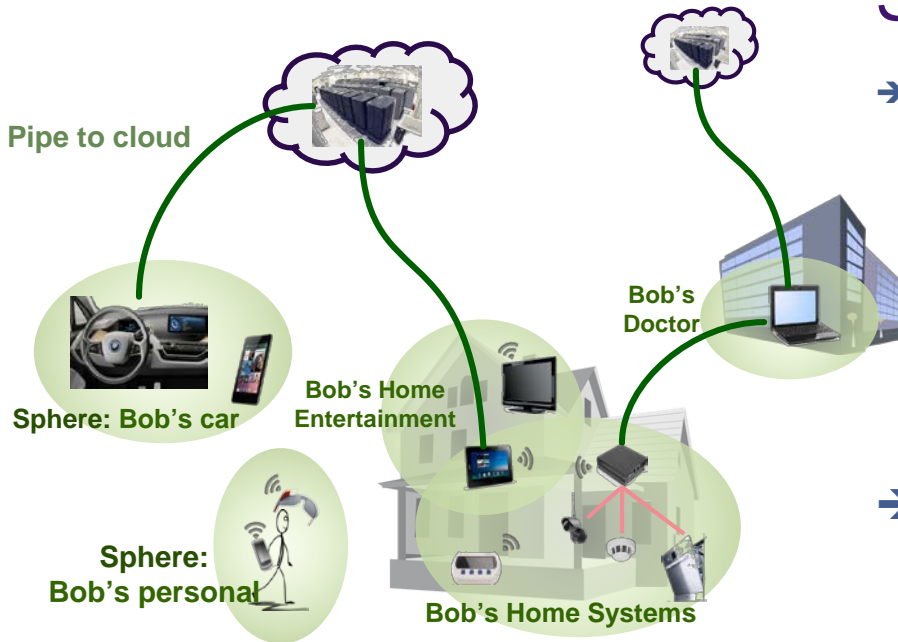
to the IoT what bio/organic is to agricultural products
IoT that tastes better



reclaim user-defined boundaries

Architecting the IoT

Hub-and-spoke → System of Systems



boundaries of confidentiality
Security model

→ **Pipes between spheres / to cloud**
secure channels for data & events

- requested by services, authorized by users
- policy enforced by middleware
only authorized exchanges go through

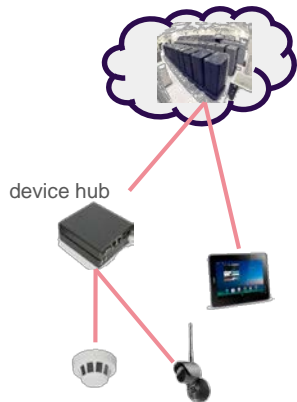
→ **Spheres of trust**

- bring security to realm of users
create sphere, join device...
- **easy** user experience
promote usability of security

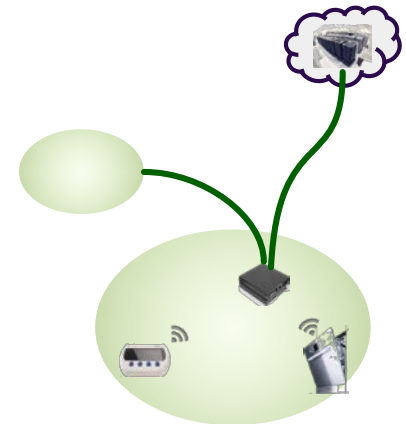
Architecting the IoT

Hub-and-spoke → System of Systems

boundaries in topology



- **Internet:** successful apps run on general purpose computers and access remote services
e.g. email, web browsing
- **IoT:** must a sensor/appliance shoulder the burden of a peer on the internet?
e.g. access control...
 - a sensor/appliance does not communicate primarily with remote services



our
claim

IoT topology should recognize and support two kinds of communication scopes: **local** and **remote**

IoT ≠ give every device an IP(v6) address

Architecting the IoT Topology ⇔ addressing a Thing

landscape of addressing schemas

address	applications	who receives
node e.g. 172.16.254.1 (IPv4)	Internet routing: IPv4 (1981), IPv6 (1998) LANs: Bluetooth, WiFi...	identified node sender must know recipient's address
geo e.g. (40.426, -79.965, 500) (lat, long, radius)	sensor networks, safety & disaster response, transportation	whoever is in the area
topic label e.g. "user location"	pub/sub: Java Messaging Service (message centric), Data Distribution Service (data centric)	whoever subscribes to the topic

network def.

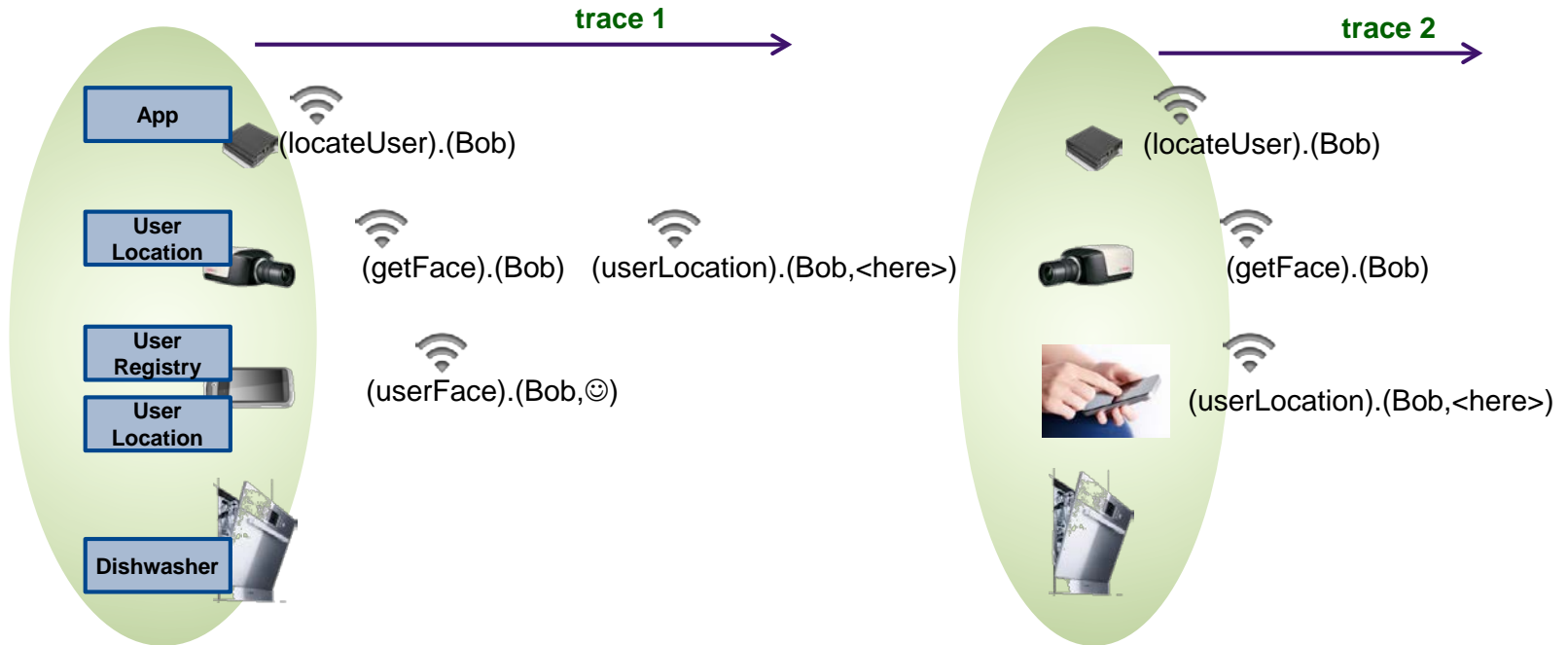
application defined

different addressing schemas solve different problems

Architectural Practice

Addressing by Intention

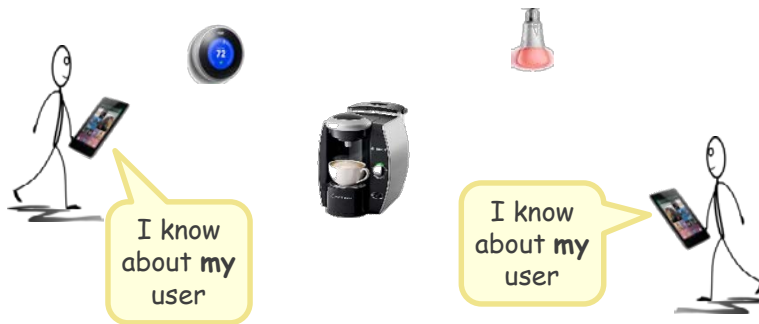
Communication within topological boundaries



- how a request is resolved depends on the **status** of the environment
- no need to scale unique internet addressing to every device

Architectural Practice

Promote **decentralized IoT**



personalized experience

everywhere

Turn lights on
Why are they red?
Get me a coffee!

Interoperation Protocols

Addressing by intention

Spheres & Pipes

brokerless pub-sub



open, multivendor

dynamic & resilient

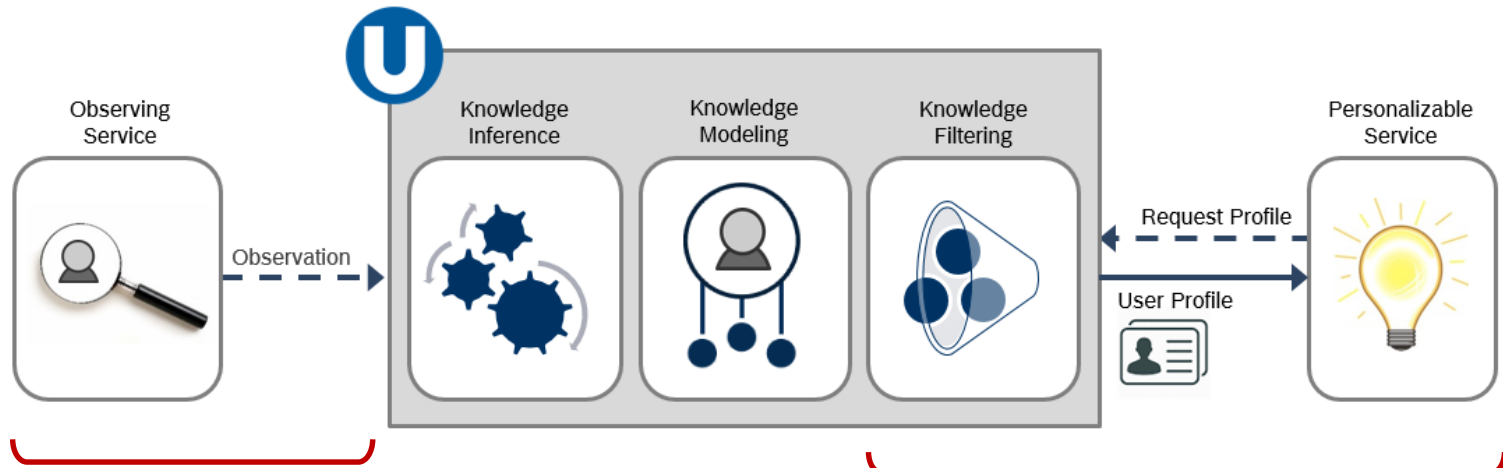
private & secure

impromptu comms.
no single-point-of failure

Architectural Practice

decentralized → emphasizes Protocols

example: learning how user engages the environment



Dragonfly



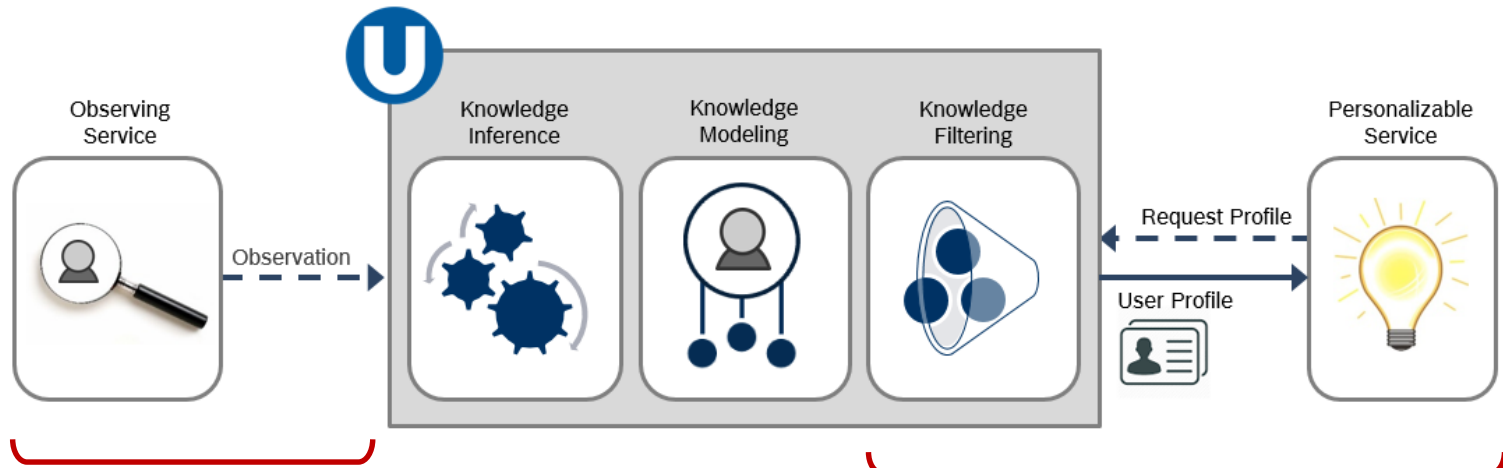
Penguin

- External service observes the user's lighting choices (e.g., yellow lights on), and the context in which the choices occurred
- External service shares these observations within the network (and with U)

Architectural Practice

decentralized → emphasize Protocols

example: tailored *user profile* upon request

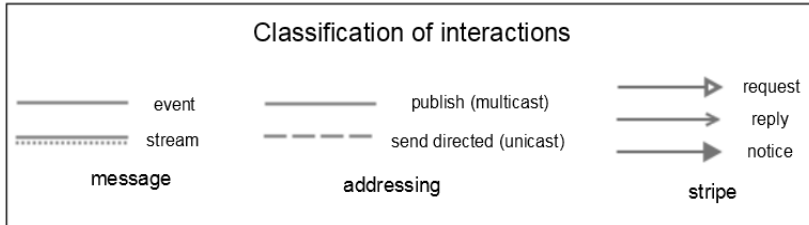


Penguin

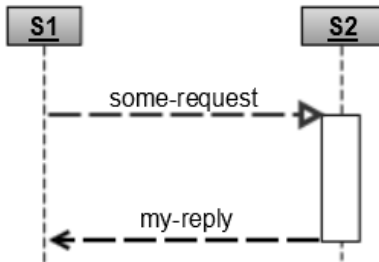


- Receive request - service requests the profile of a user
- Filter knowledge - U decides what user knowledge is relevant for the service.
- Encode knowledge - U encodes the relevant knowledge into a profile and sends to the requesting service

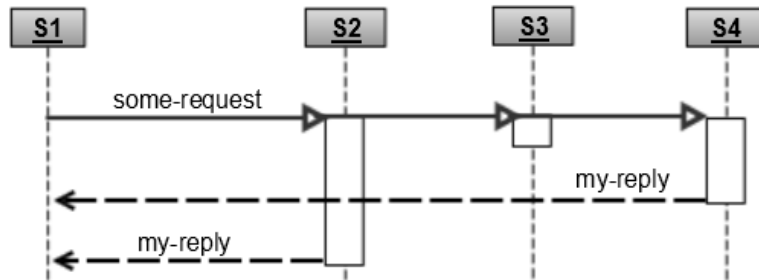
Addr. by Intention → Rich forms of request-reply



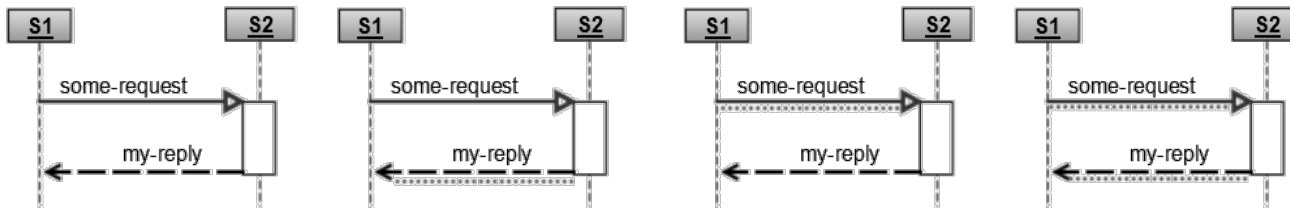
Classic request-reply



Group request-reply

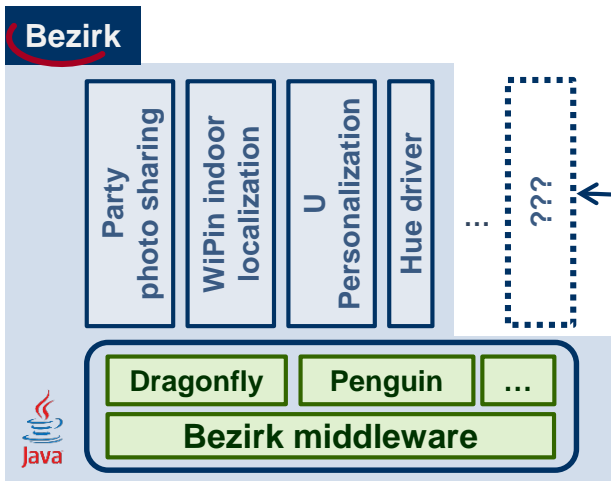
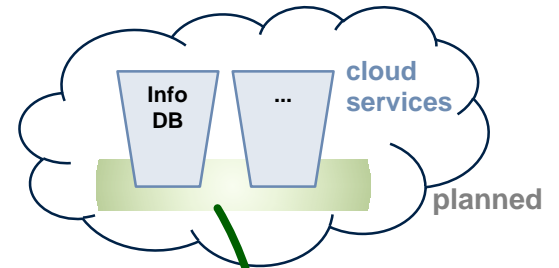
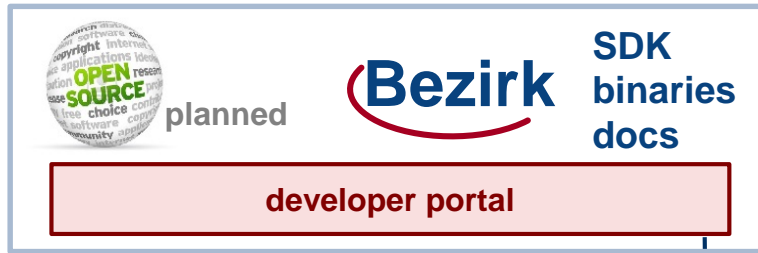


Combinations of event/stream request-reply



Open developers' community

<http://www.bezirk.com>



① you may download

- end-users:**
- access cloud services
 - up/download content

③ try out new services & use cases

② code over

- services / apps
- protocols

④