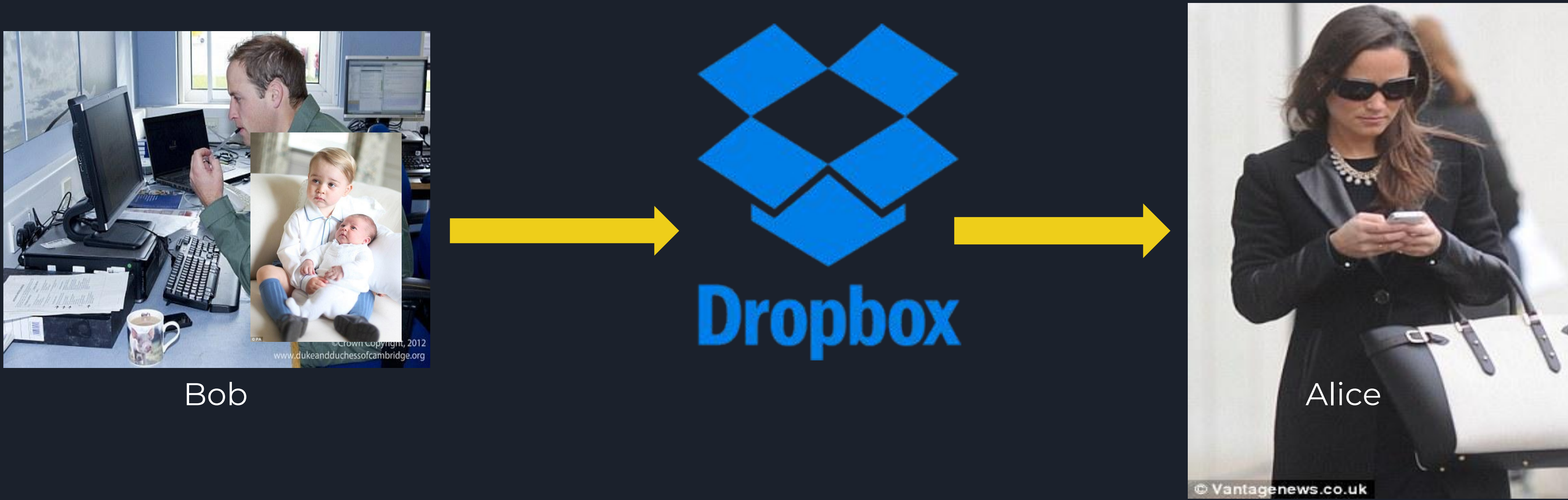
Portable Data Integrity and Confidentiality using Graduated Access Control

Puneet Mehrotra, Amanda Carbonari, Peter Chen, Ivan Beschastnikh, Andrew Warfield

Network, Systems, and Security Lab
CS@UBC

Motivating Example



©Crown Copyright, 2012
www.dukeandduchessofcambridge.org

© Vantagenews.co.uk

Bob

Dropbox

Alice

What could go wrong?

Dropbox gets compromised

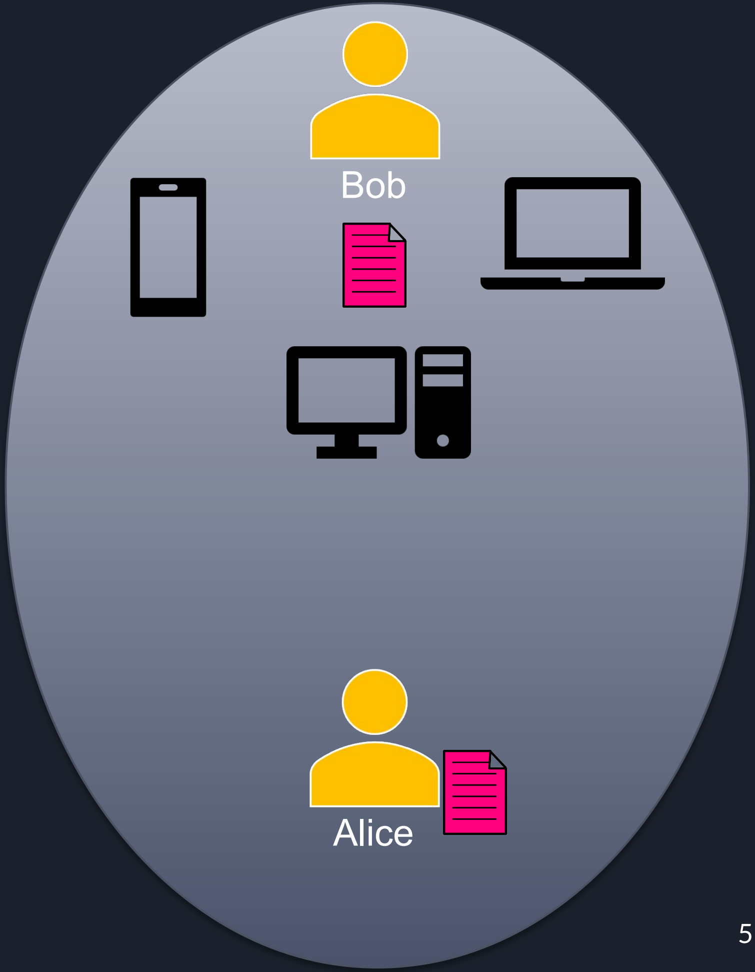
Curious eyes in public spaces

Changed circumstances –
lost/stolen device

Bob’s data is mobile, Bob’s data access policy is **NOT**

Problem 1: distribution of the data and the data access policy are synonymous and binary

Problem 2: data access policy on remote devices (e.g., Alice’s phone) may be inadequate or not enforced

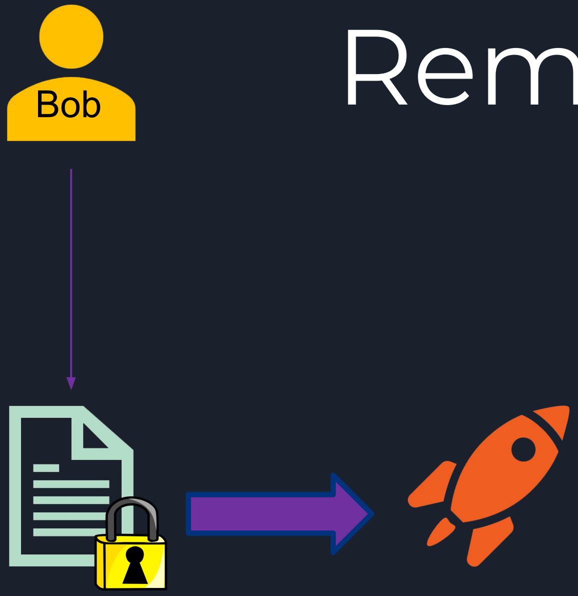


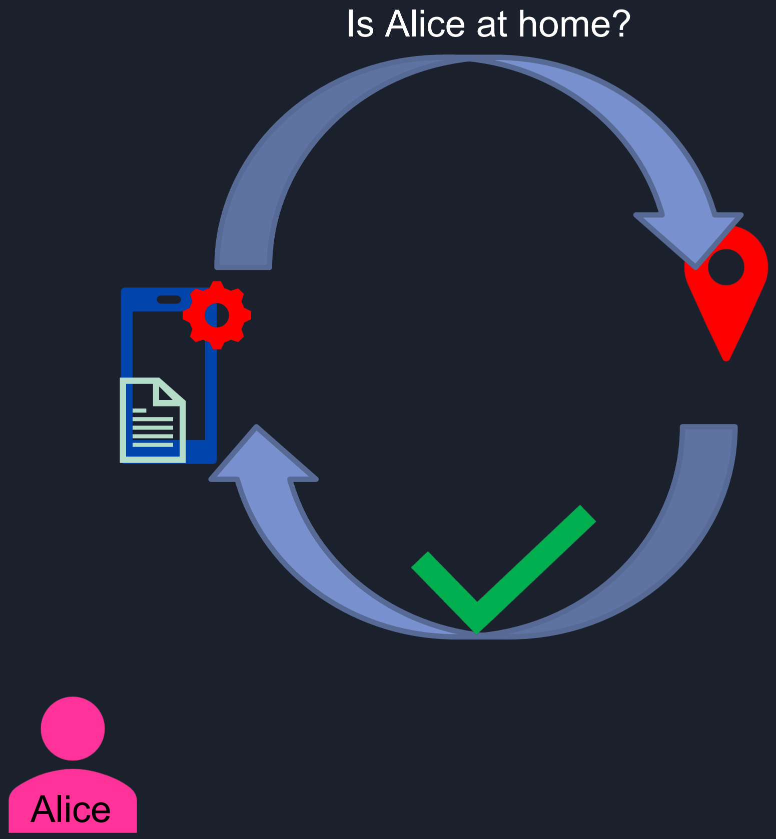
Access Policies Today

- Depend only on the application and device it is on
- Evaluated at time of distribution
- Binary decision at time of sharing (can/cannot)

Enforceable access policies for mobile data

Graduated Access Control on Remote Devices





Mobile
Dynamically Resolvable
Programmable
Backward Compatible

✓

Benefits

Dropbox gets compromised

Data remains safe and encrypted

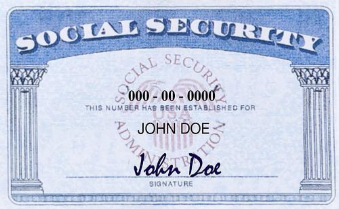
Curious eyes in public spaces

Bob decides ***who*** can access his data and ***where***

Changed circumstances – lost/stolen device

Bob can remotely revoke access to his device

Usecases



SOCIAL SECURITY

000 - 00 - 0000

THIS NUMBER HAS BEEN ESTABLISHED FOR

JOHN DOE

John Doe

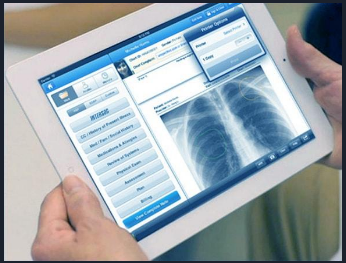
SIGNATURE

Identity Protection

Revocation: remote delete, auto delete, policy change, remote state change

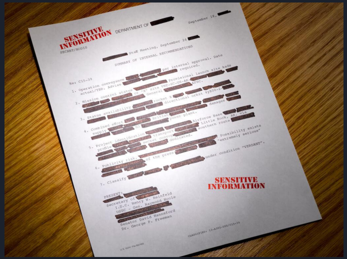
Data integrity and Provenance

Detect tampered data, audit trails



Electronic Health Records

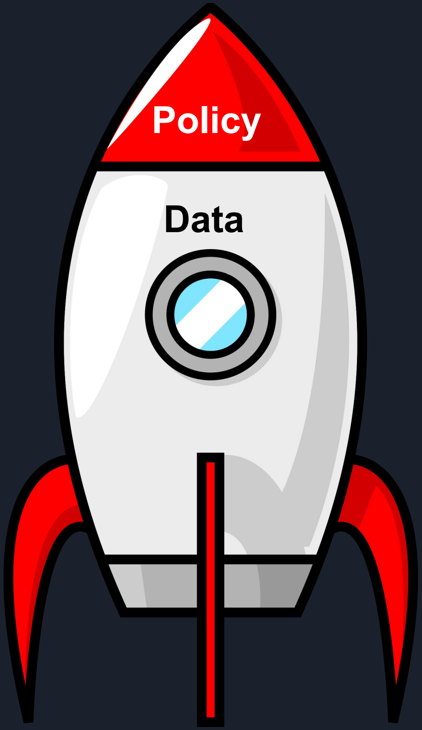
Audit trails, role-based access



Sensitive Documents

Redaction, geo-fencing, time-fencing, role-based access

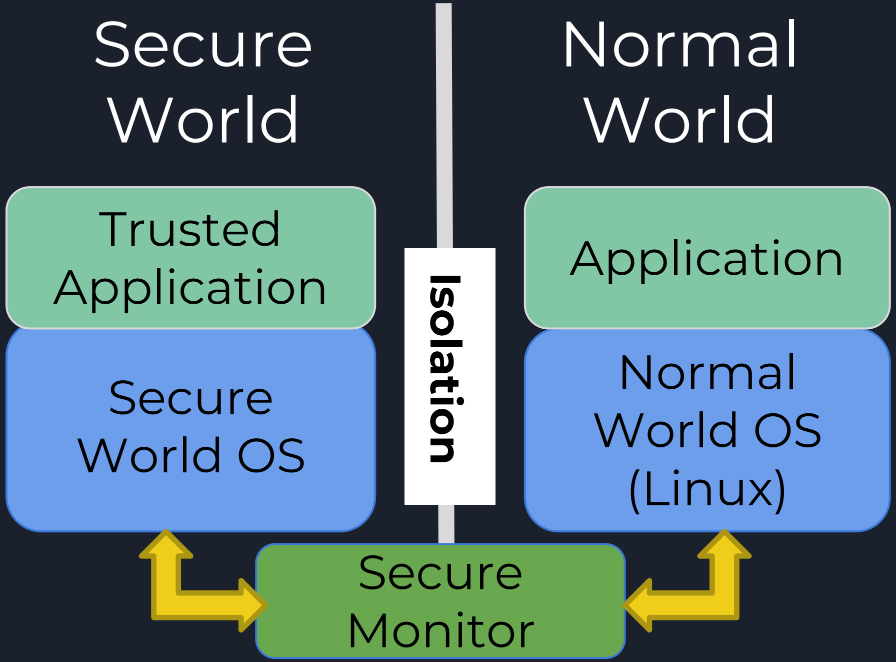
Our Solution



- Data centric abstraction of graduated access control

- Data and Policy encrypted together in a single mobile unit : **Trusted Capsule**

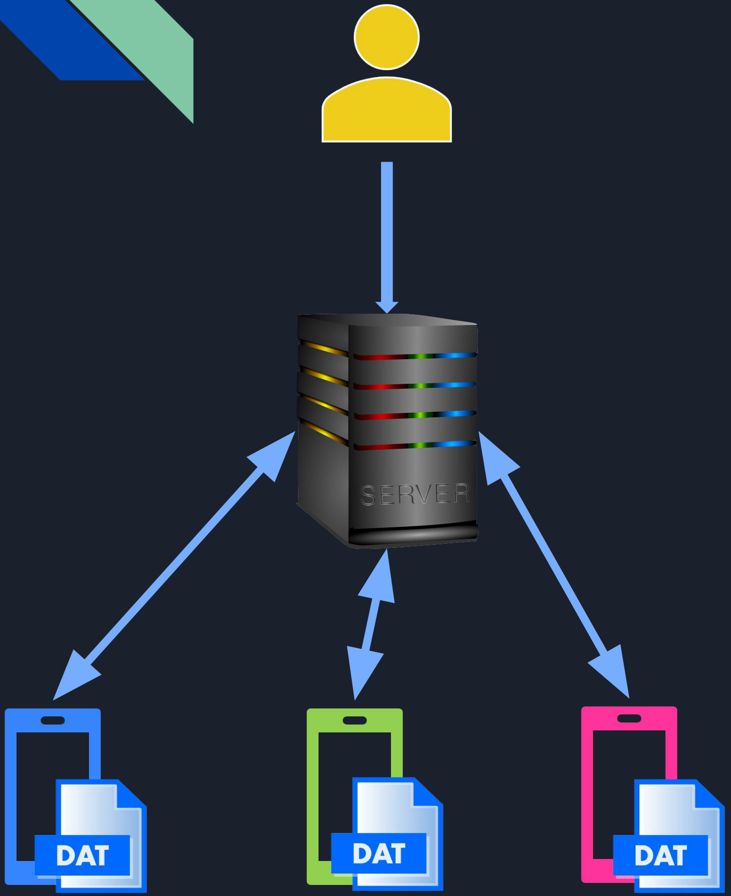
Our Solution : Trusted Execution Environment



- Examples include Intel Secure Guard eXtensions (SGX) and ARM TrustZone
- Available on commodity ARM chipsets
- Hardware partitions CPU and memory into two logical worlds: secure and normal
- Compromising Normal World does not compromise Secure World

- Implement system call interceptor to allow applications to transparently operate on trusted capsules
- Secure world offers TEE for trusted capsule applications to evaluate capsule policy at syscall granularity

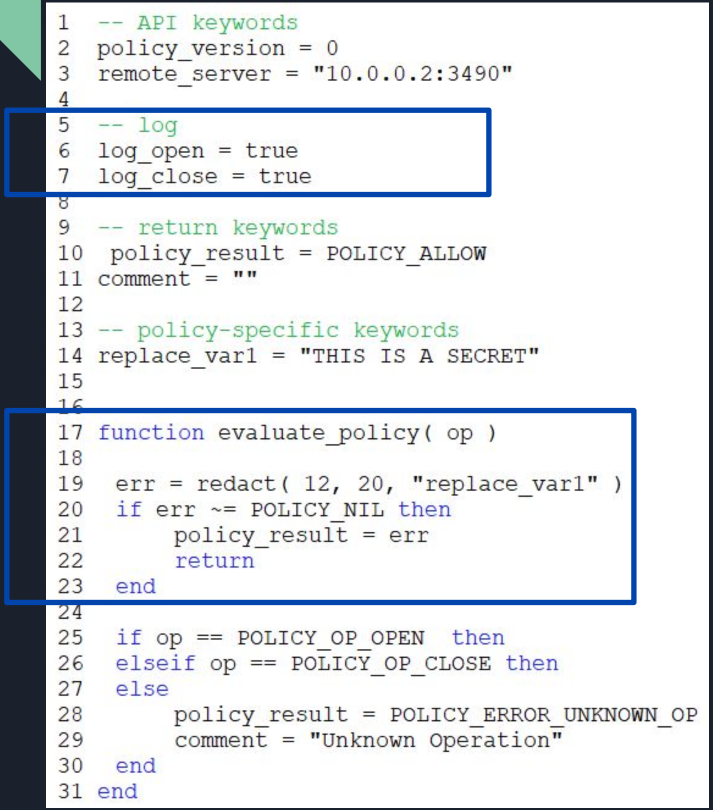
Our Solution: Trusted Capsule Server



- Maintain data owner policy uniform across all trusted capsule copies

- Actions:
    - Receive logging information from trusted capsules
    - Initiate policy change (ex: Remote delete)

Our Solution: Policy Engine



```
-- API keywords
policy_version = 0
remote_server = "10.0.0.2:3490"

-- log
log_open = true
log_close = true

-- return keywords
 policy_result = POLICY_ALLOW
comment = ""

-- policy-specific keywords
replace_var1 = "THIS IS A SECRET"


function evaluate_policy( op )

  err = redact( 12, 20, "replace_var1" )
  if err ~= POLICY_NIL then
        policy_result = err
        return
  end

  if op == POLICY_OP_OPEN  then
  elseif op == POLICY_OP_CLOSE then
  else
        policy_result = POLICY_ERROR_UNKNOWN_OP
        comment = "Unknown Operation"
  end
end
```

- Lua based policy language
- Global variables – trusted server IP and port
- States
  - Normal world OS states ex: process ids
  - Peripheral device information
  - Remote states
- Evaluates policy on *op* where op is the system call

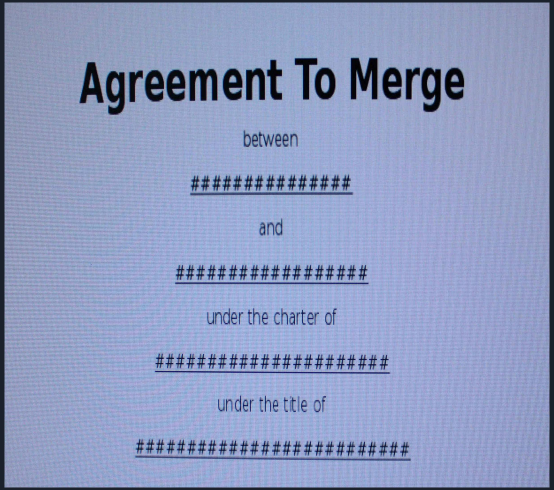
Implementation

Samsung Knox uses ARM TrustZone

- Prototype on LeMaker HiKey
  - ARM Cortex A53 processors
  - 8 GB eMMC Flash
  - 2 GB RAM
  - TrustZone unlocked
- Linaro OP-TEE OS version 1.0 (Secure World)
- Debian Linux Kernel 3.18.0 (Normal World)
- 128-bit AES and SHA-256 (Trusted Capsules)

Evaluation – Policy Language

Policy	LOC
Merger Document	24
Transcript	25
Royal Photo	30
EHR	41



Agreement To Merge

between

###############

and

##################

under the charter of

######################

under the title of

##########################

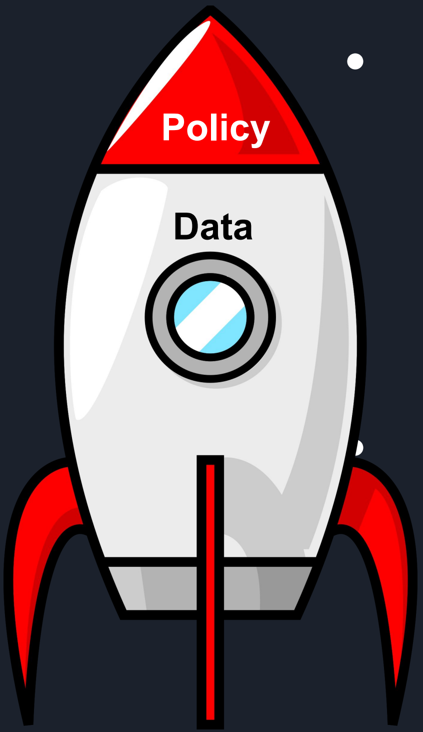
- Express all our use case policies with small LOCs
- Complex policies such as redaction can be expressed with few lines of code
- Lua interpreter required <2KB of stack

Evaluation – Storage Overhead

	Data (KB)	**Capsule (KB)**	**Overhead (%)**
PDF Doc	137.34 KB	139.38 KB	1.42%
JPEG Image	204.10 KB	207.00 KB	1.42%
MP4 Video	4142.40 KB	4175.94 KB	0.80%
FODT Doc	54.80 KB	56.70 KB	3.47%

- Negligible storage overhead

Conclusion



- Current day policies are application/device-centric, evaluated once, binary and unchangeable
- We introduce **graduated access control**
  - Data owner can enforce access policies on remote devices
  - Define a continuum of actions rather than a binary can/cannot
  - Decouples access policy from data distribution
- **Trusted capsules** based implementation using ARM TrustZone as our TEE
  - Mobile
  - Dynamically Resolvable
  - Programmable

Backup Slides

Graduated Access Control on Remote Devices

- **Mobile** data access policy moves with the data

- **Dynamically Resolvable** data access policy re-evaluated at time of access

- **Programmable:** data access policy is nuanced

- **Backward Compatible:** does not require application modification