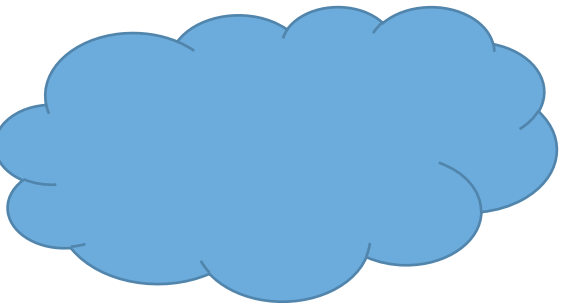
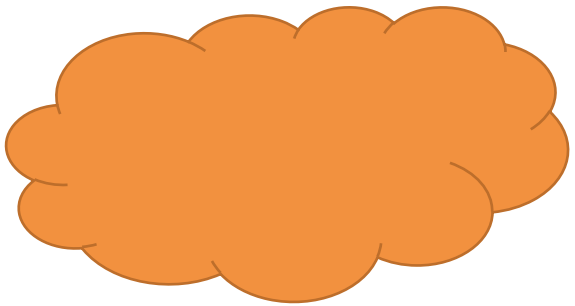
Distributed System Security in a Dynamic Environment

PW (with Acknowledgements to Mark Little)

Deployment Today

Networks





Issues:

BYOD

Mobile Networ

Roaming

Dynamic Deployment

Concerns

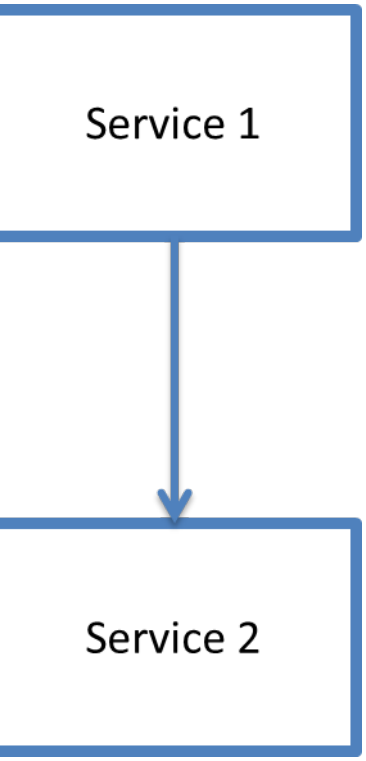
- Is it safe to deploy the client to an application on an unknown BYOD?
- Is safe for corporate data to be transferred to and from mobile devices over home broadband, coffee-shop Wi-Fi and phone networks?
- Which components can be safely deployed on a public cloud?
- Which data items can be safely transferred over the Internet to a public cloud?

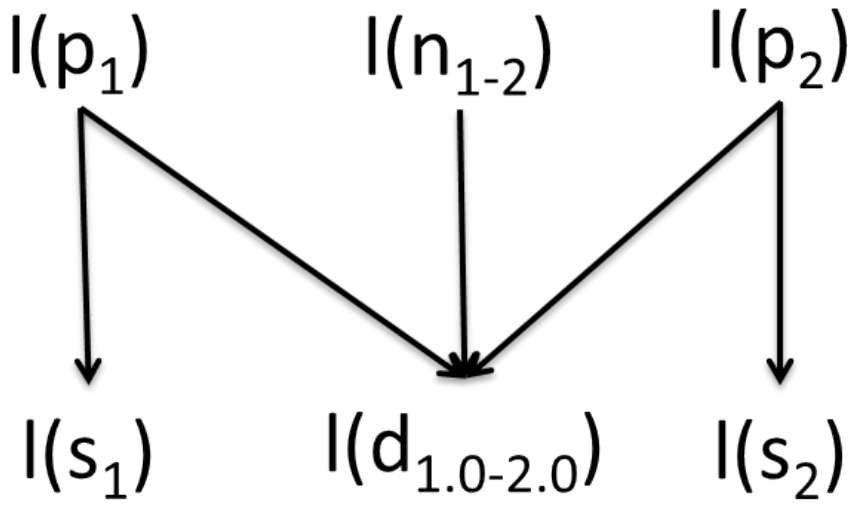
Aim: a formal approach to answering these questions

- Administrator specifies required security levels for services & data
- Checking
  - Administrator specifies
    - platform on which each service is to be deployed
    - networks it utilises
  - The method determines if security requirements will be met
- Exploration
  - Administrator specifies
    - range of platforms on which each service could be deployed
  - The method generates all deployment options that meet the requirements

Method

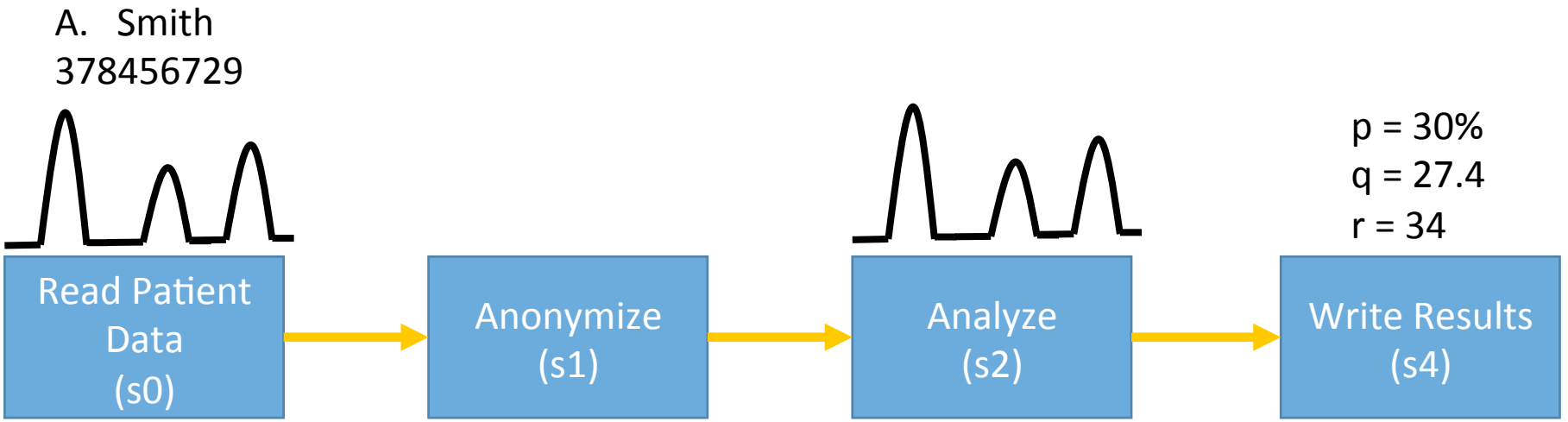
- Represent application as a directed graph
- Model Platform, Network, Service, Data





$l(p1) \geq l(s1) \wedge$
$l(p2) \geq l(s2) \wedge$
$l(p1) \geq l(d1.0{-}2.0) \wedge$
$l(p2) \geq l(d1.0{-}2.0) \wedge$
$l(n1{-}2) \geq l(d1.0{-}2.0)$

Example 1: Healthcare Data Analysis



Step 1: Create set of Inequalities

$l(p0) \geq l(s0) \wedge$
$l(p1) \geq l(s1) \wedge$
$l(p2) \geq l(s2) \wedge$
$l(p3) \geq l(s3) \wedge$

$l(p0) \geq l(d0.0{-}1.0) \wedge$
$l(p1) \geq l(d0.0{-}1.0) \wedge$
$l(n0{-}1) \geq l(d0.0{-}1.0) \wedge$
$l(p1) \geq l(d1.0{-}2.0) \wedge$
$l(p2) \geq l(d1.0{-}2.0) \wedge$

$l(n1{-}2) \geq l(d1.0{-}2.0) \wedge$
$l(p2) \geq l(d2.0{-}3.0) \wedge$
$l(p3) \geq l(d2.0{-}3.0) \wedge$
$l(n2{-}3) \geq l(d2.0{-}3.0)$

Example 1: Medical Data Analysis

Step 2: Where there are variables in the inequalities that represent real-world entities whose security levels are known and fixed, bind those variables to the known security levels.

l(s0)	1
c(s0)	1
l(s1)	0
c(s1)	1
l(s2)	0
c(s2)	0
l(s3)	0
c(s3)	1

l(d0.0–1.0)	1
l(d1.0–2.0)	0
l(d2.0–3.0)	0
l(n0–1)	1
l(n1–2)	1
l(n2–3)	1

Example 1: Medical Data Analysis

Step 3. Simplify the resulting set of inequalities.

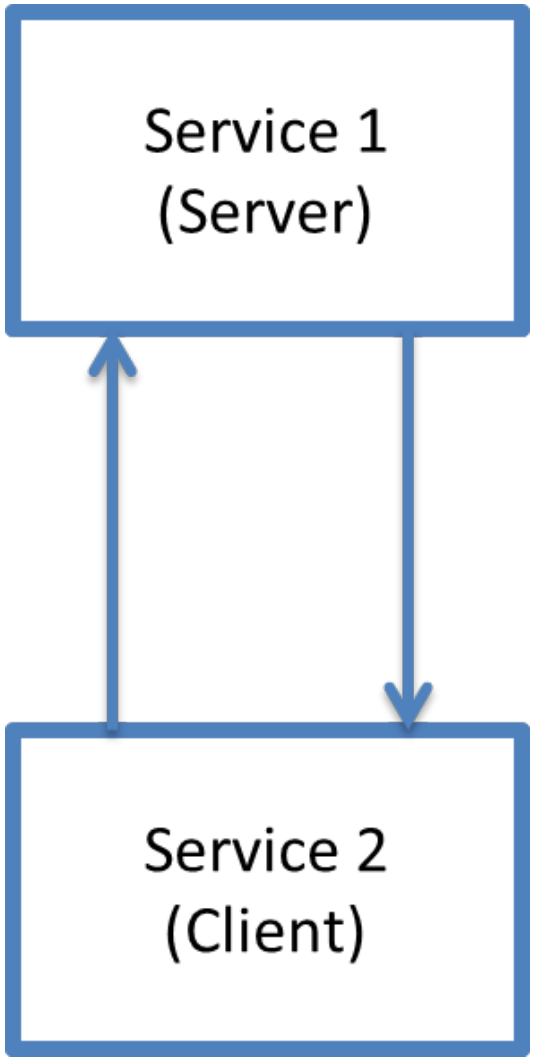
This can generate one of three results:

- The security constraints can be met
- The security constraints can not be met
- There are specific values (or ranges of values) that the unbound variables can take that would allow the security constraints to be met

$l(p0) \geq 1 \wedge$
$l(p1) \geq 1 \wedge$
$l(p2) \geq 0 \wedge$
$l(p3) \geq 0$

Examples: Client Server

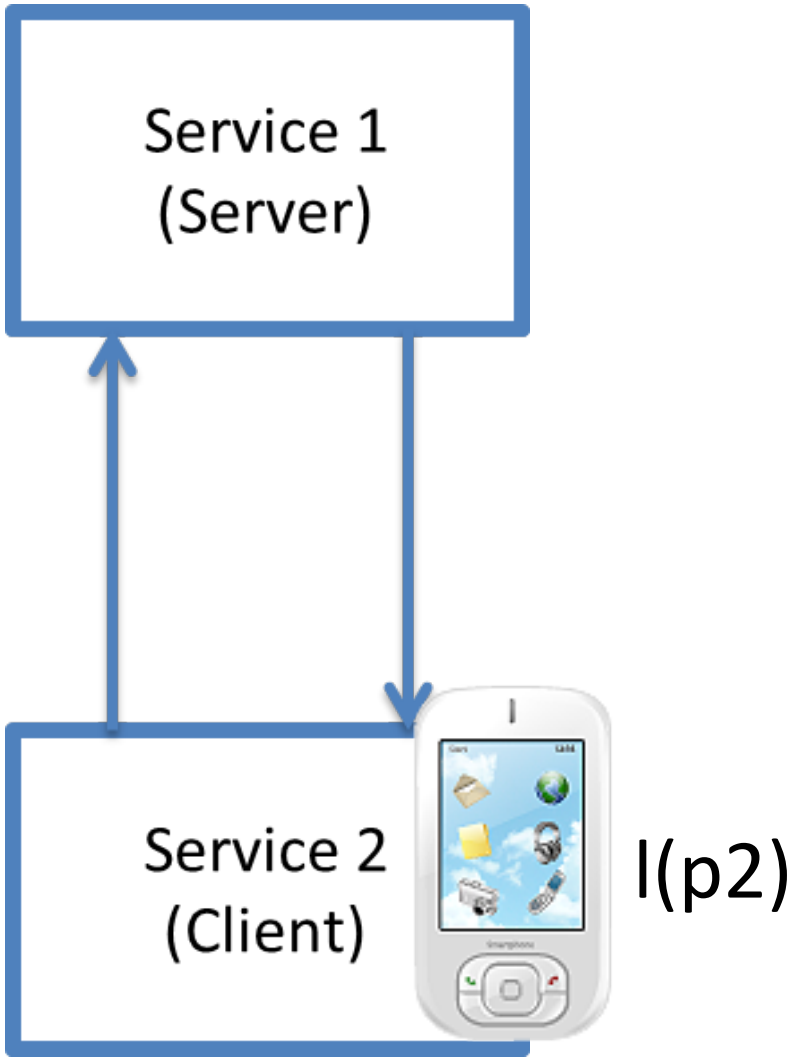
$$
\begin{aligned}
&l(p1) \geq l(s1) \wedge \\
&l(p2) \geq l(s2) \wedge \\
&l(p2) \geq l(d2.0{-}1.0) \wedge \\
&l(p1) \geq l(d2.0{-}1.0) \wedge \\
&l(n1{-}2) \geq l(d2.0{-}1.0) \wedge \\
&l(p1) \geq l(d1.1{-}2.1) \wedge \\
&l(p2) \geq l(d1.1{-}2.1) \wedge \\
&l(n1{-}2) \geq l(d1.1{-}2.1)
\end{aligned}
$$



Example: BYOD

l(p1)	1
l(s1)	1
l(s2)	0
l(d2.0−1.0)	1
l(d1.1−2.1)	1
l(n1−2)	1
l(p2)	?

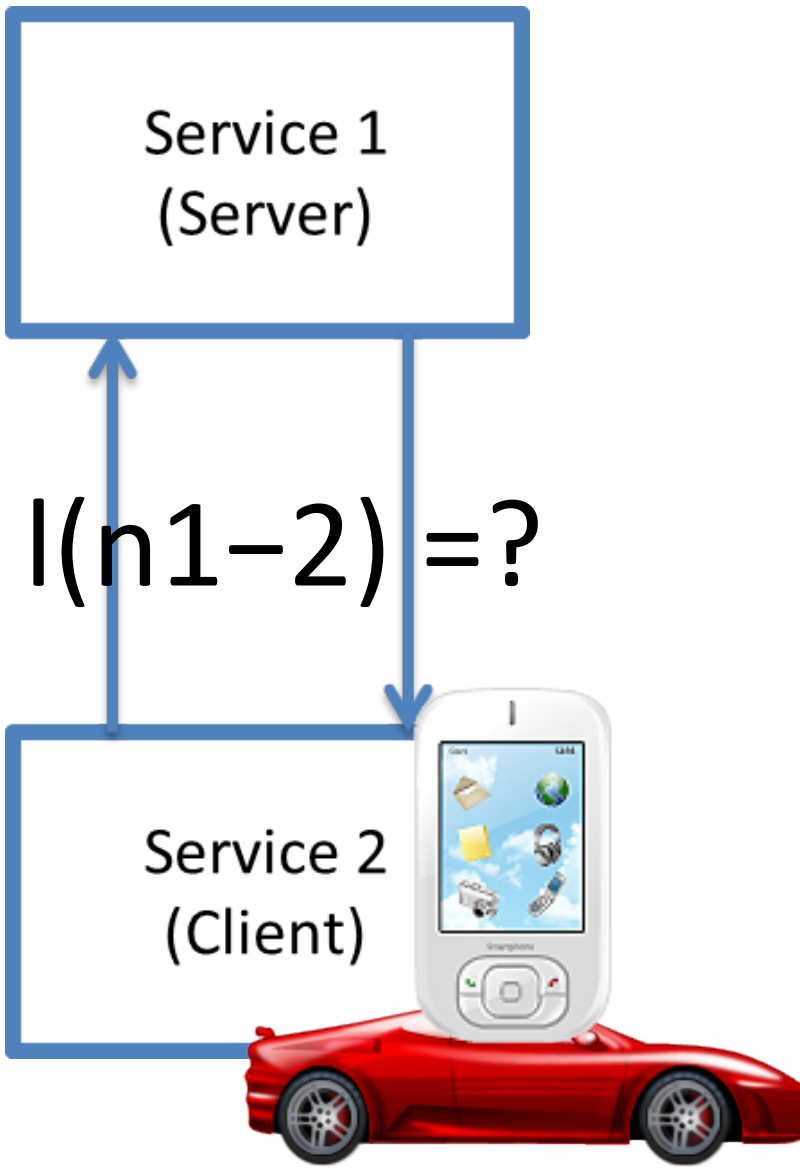
→ l(p2) ≥ 1



Example: Roaming

l(p1)	1
l(s1)	1
l(s2)	0
l(d2.0−1.0)	1
l(d1.1−2.1)	1
l(p2)	1
l(n1−2)	?

→ l(n1-2) ≥ 1



A Systematic Approach to Cloud Federation