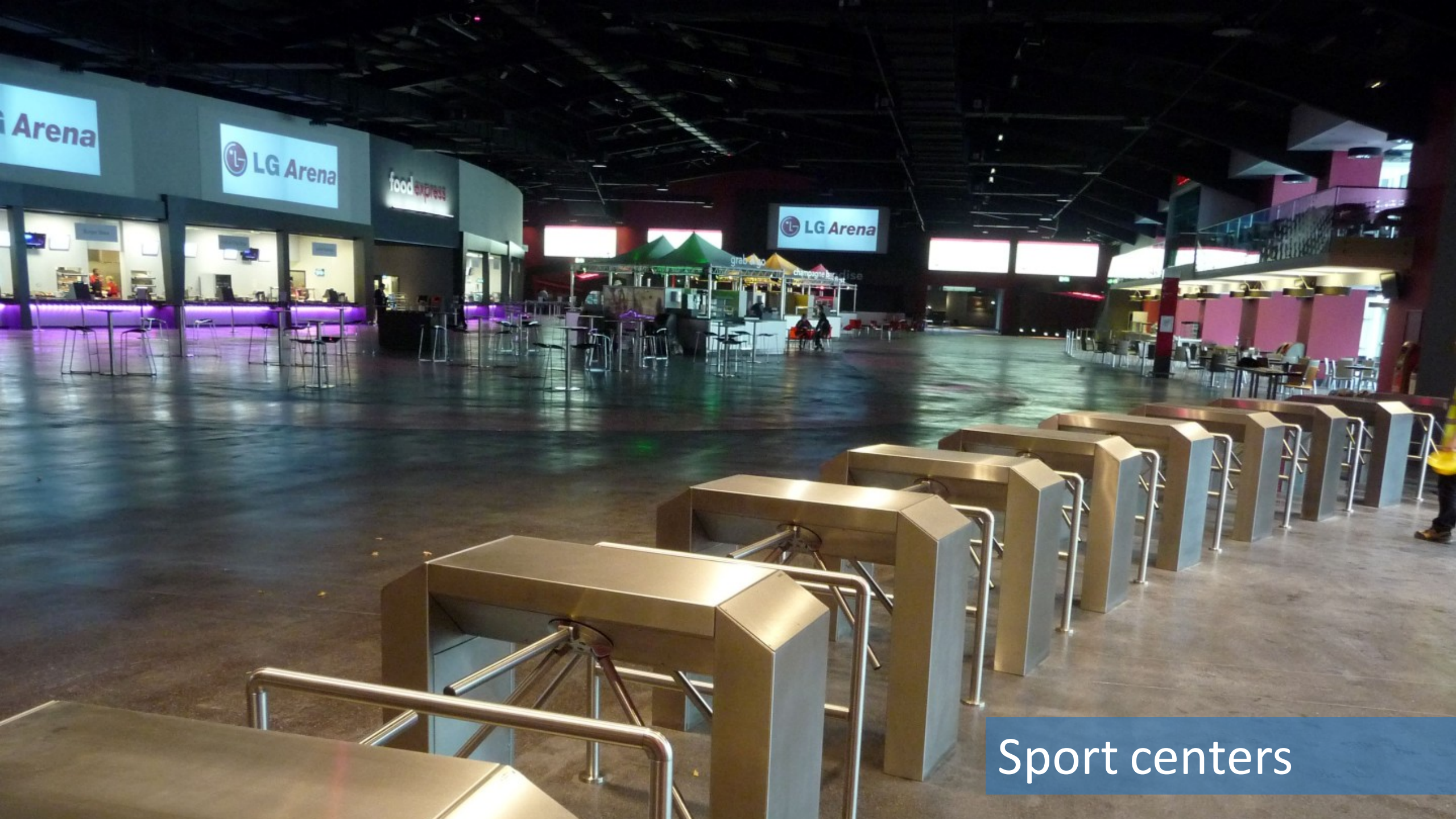
Access Control Synthesis for Physical Spaces

Petar Tsankov, Mohammad Torabi Dashti, David Basin

Institute of Information Security, ETH Zurich

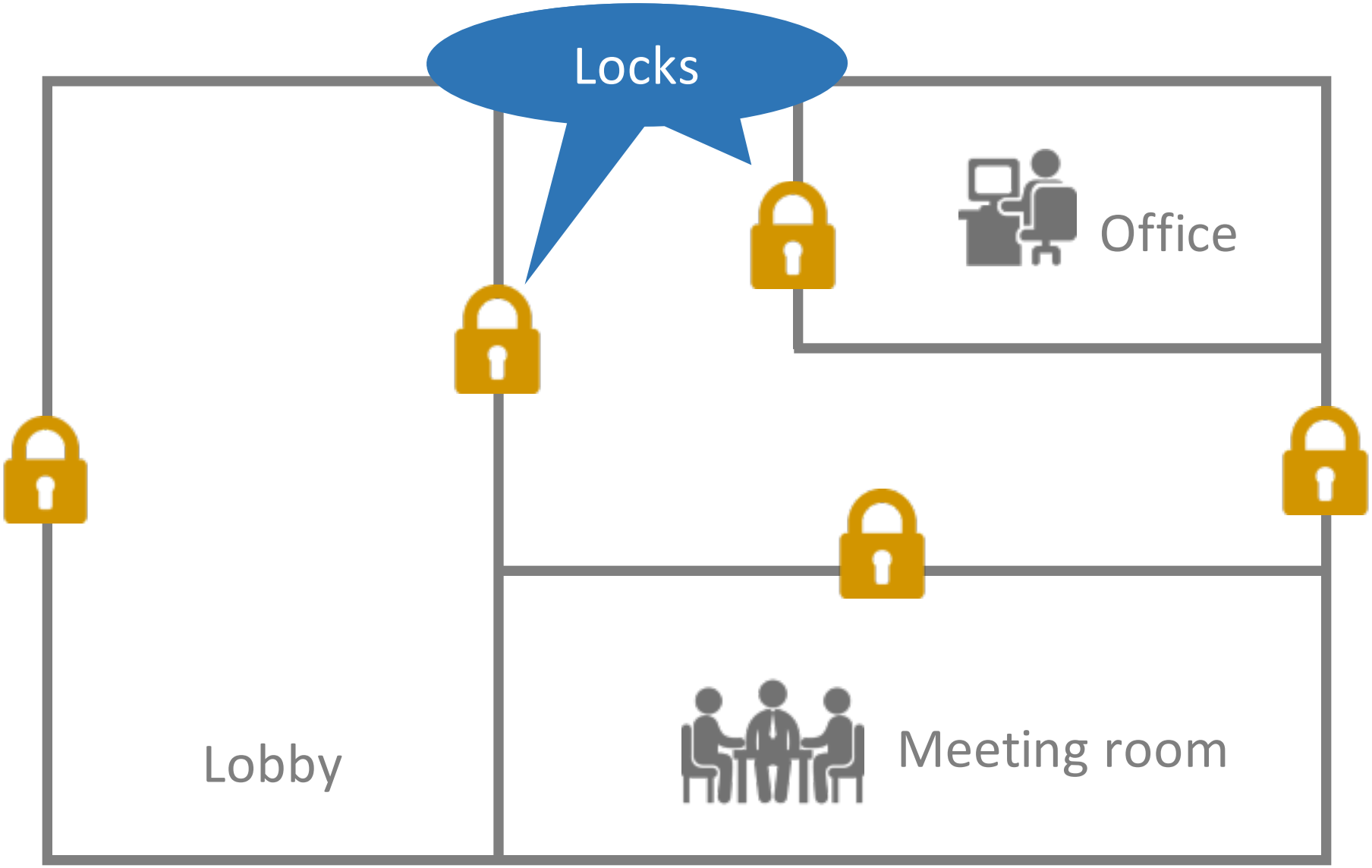
Airports


Corporate buildings

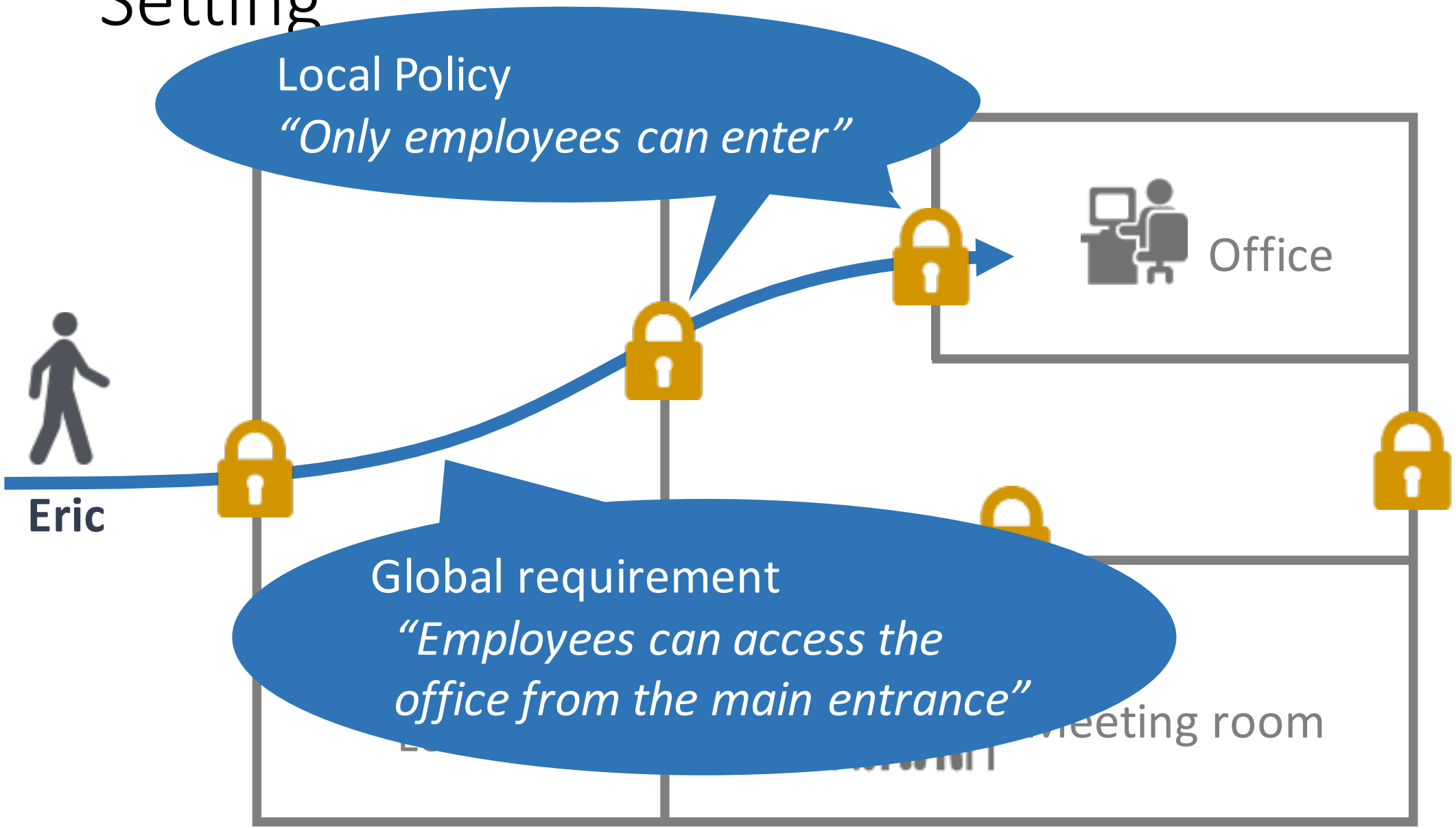


Sport centers

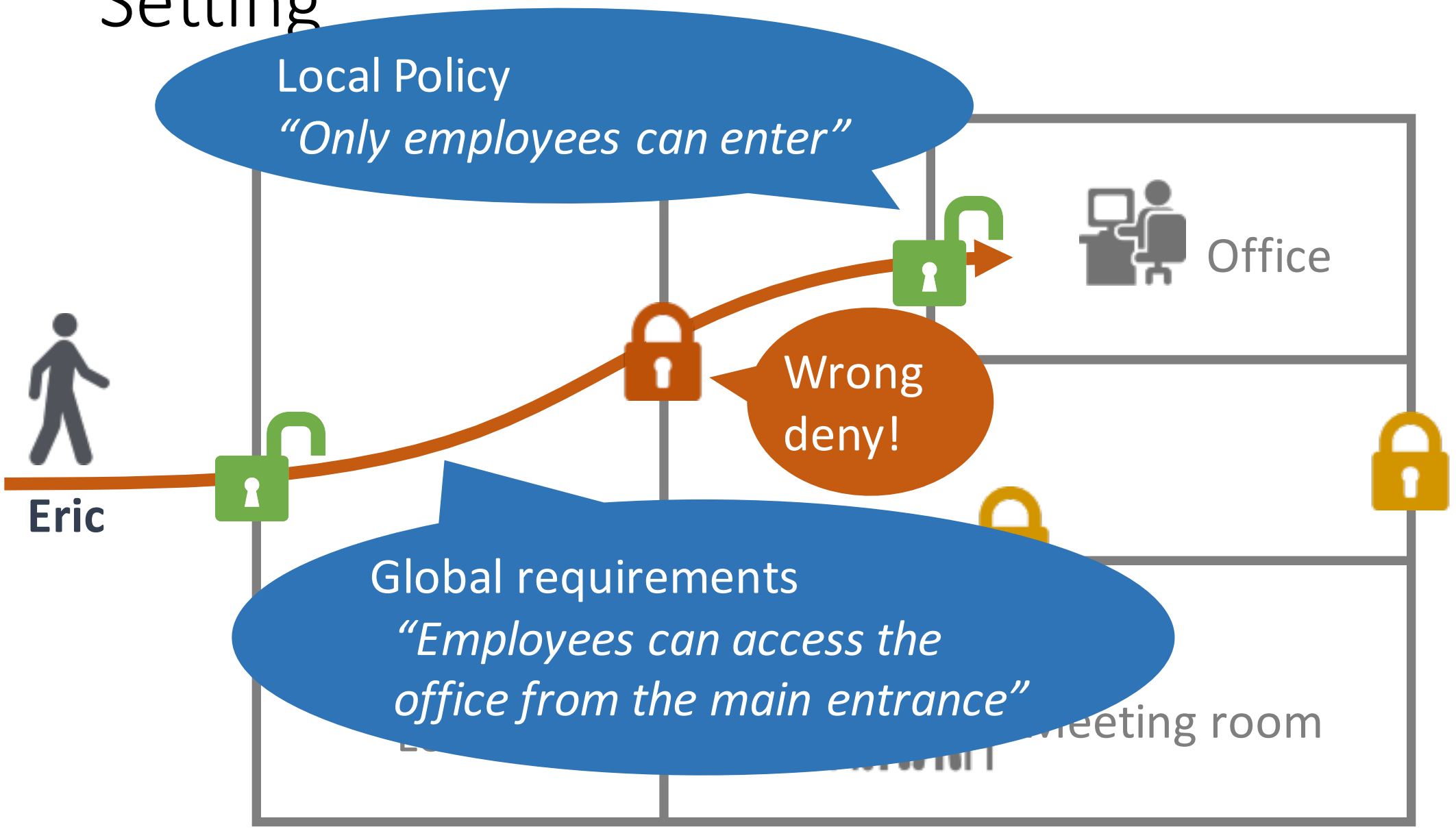
Setting



Setting


Local Policy
"Only employees can enter"
Office
Eric
Global requirement
"Employees can access the office from the main entrance"
eeting room

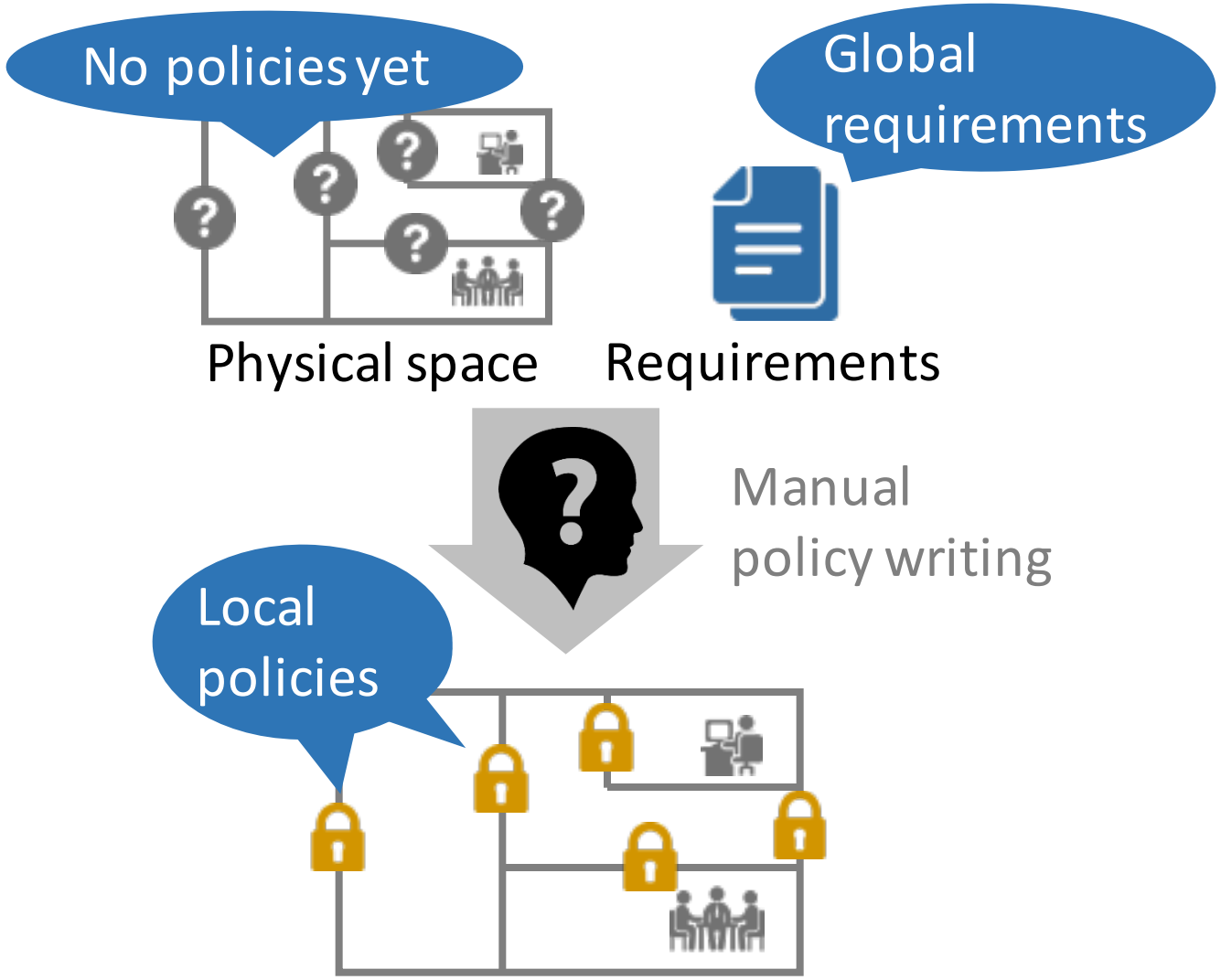
Setting


Local Policy
"Only employees can enter"
Office
Wrong
deny!
Eric
Global requirements
"Employees can access the office from the main entrance"
eeting room

Setting


Local Policy
“Only employees can enter”
Challenge
Come up with local policies that enforce all global requirements
Eric
“Employees can access the office from the main entrance”
ffice
Meeting room

Current Practice



Problems

Cannot satisfy requirements one-by-one

Cu

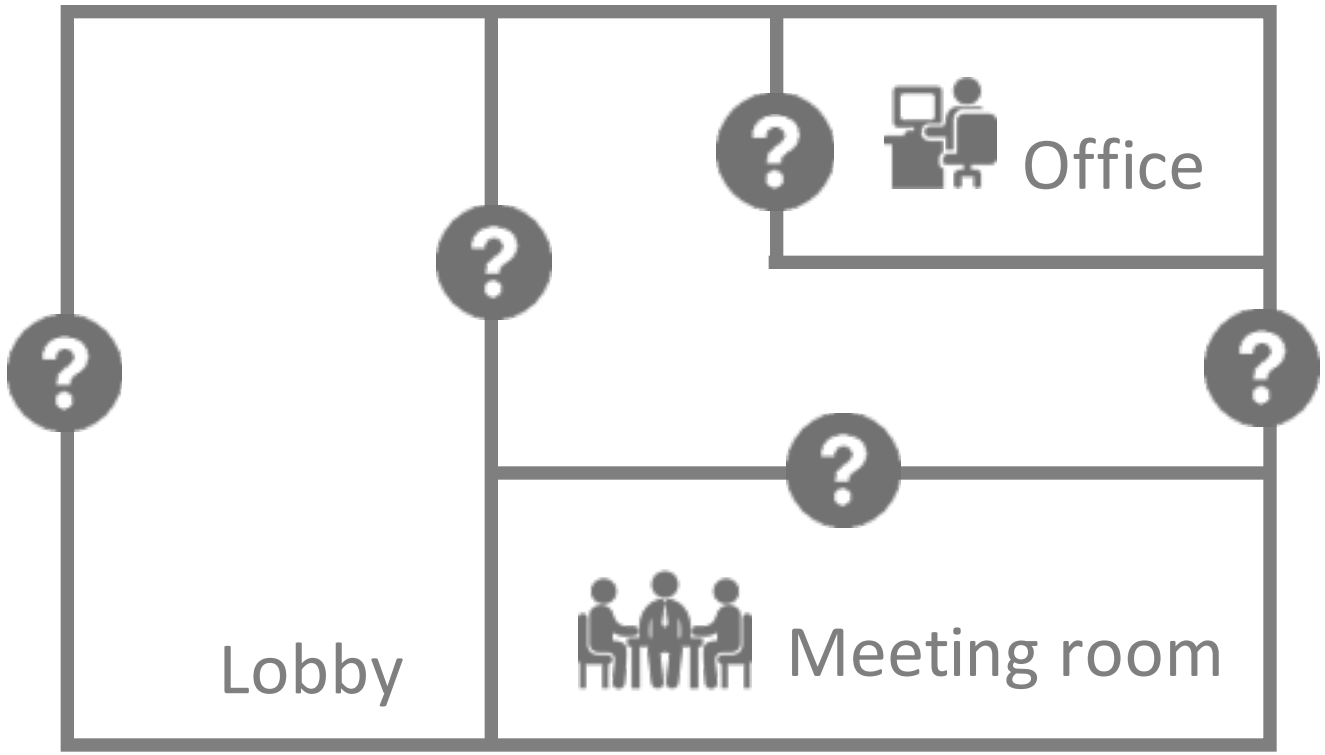
No polici

?

Phy

Loca
poli

Example



R1: Visitors can access the meeting room

y-one

Cu

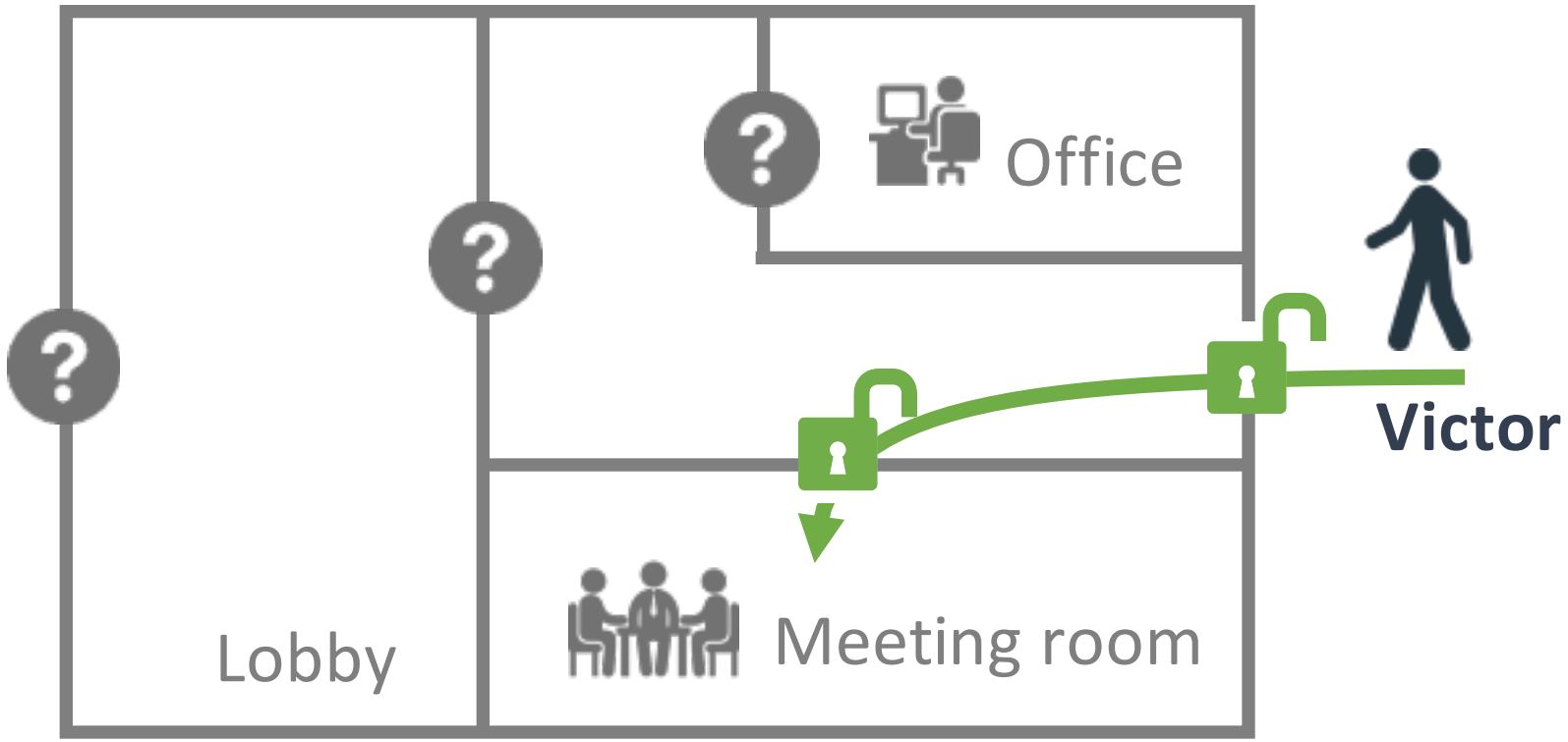
No polici

Phy

Loca
poli

y-one

Example



R1: Visitors can access the meeting room

Cu

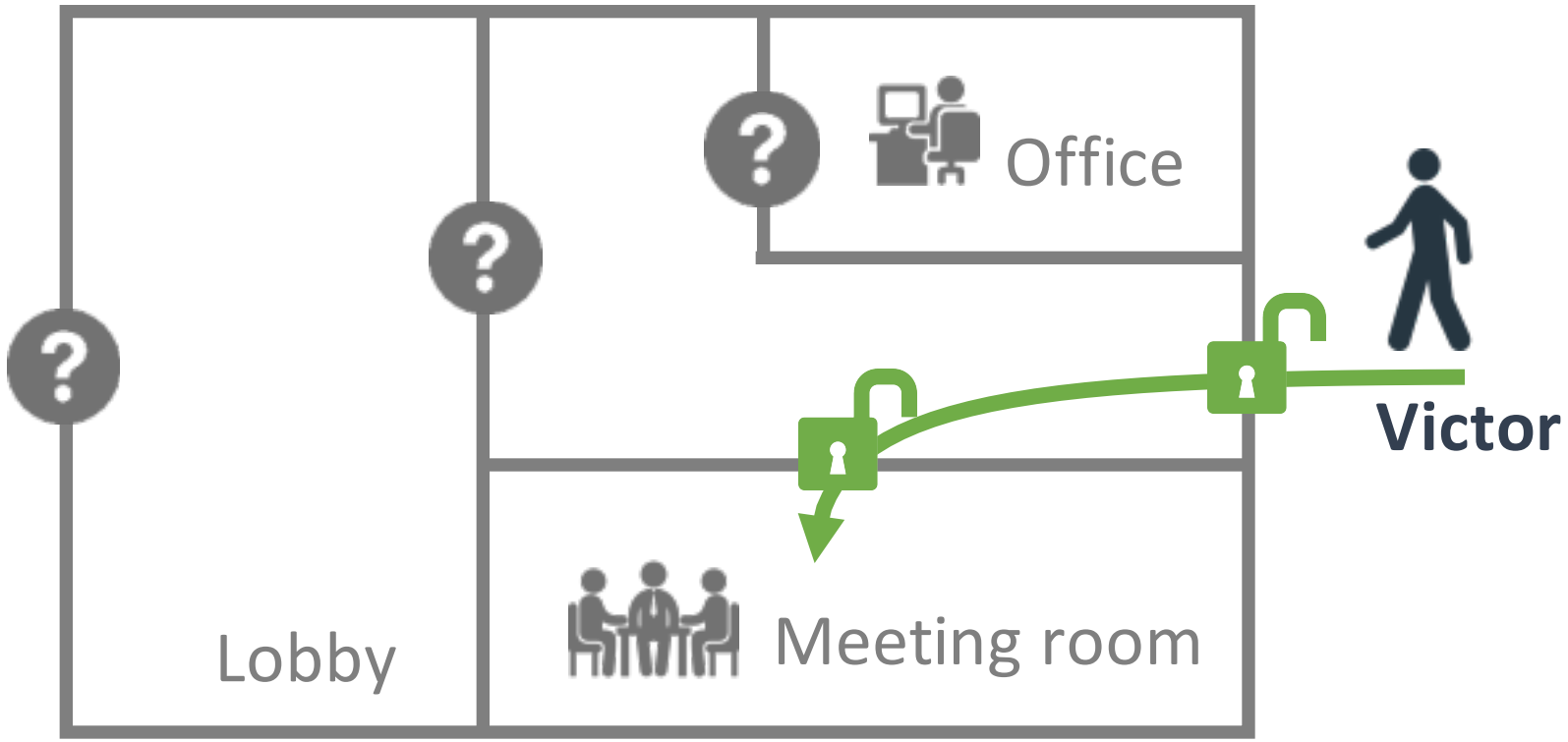
No polici

Phy

Loca
poli

y-one

Example



✔ **R1: Visitors can access the meeting room**

R2: Visitors cannot access the meeting room if they have not passed through the lobby

Cu

No polici

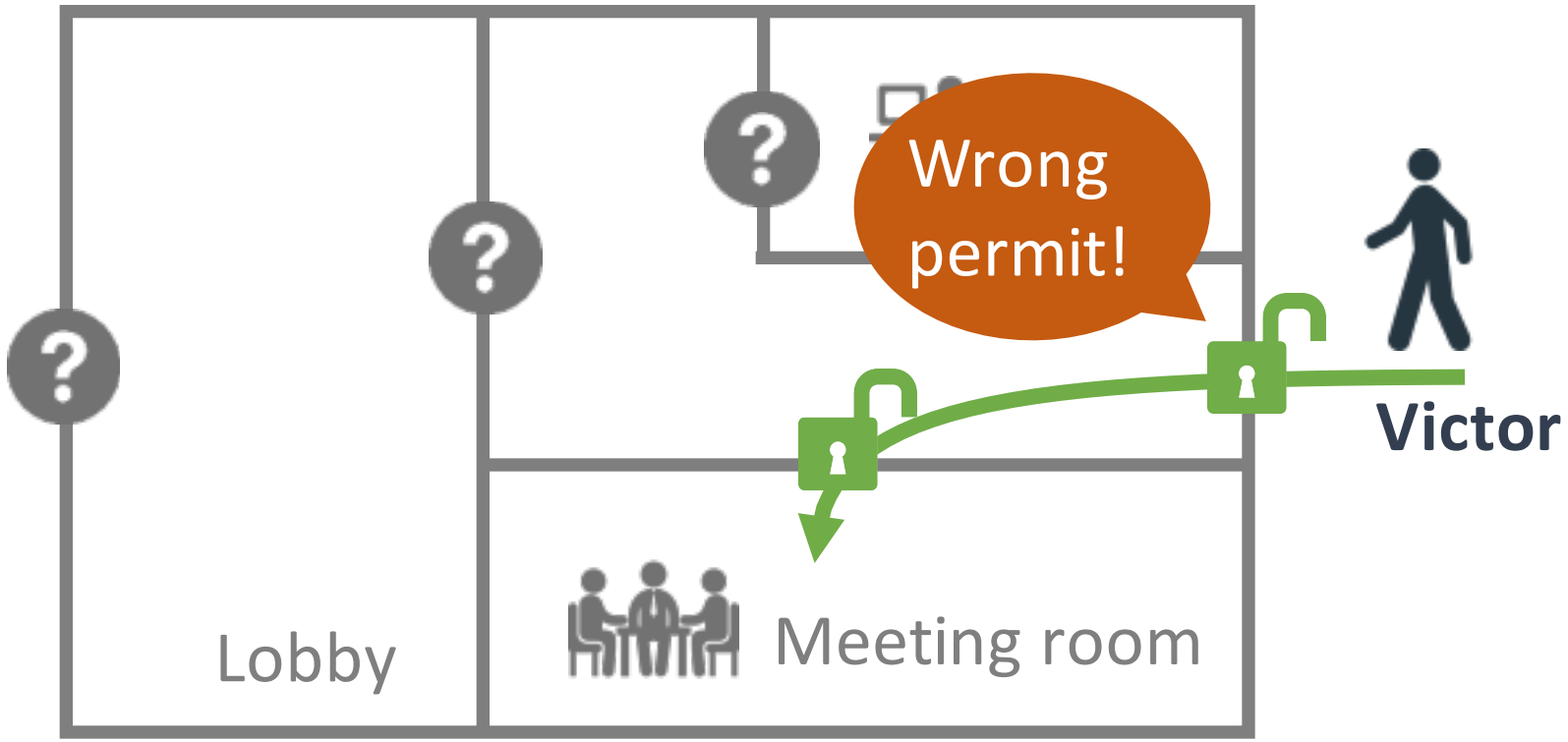
?

Phy

Loca
poli

y-one

Example



R1: Visitors can access the meeting room

R2: Visitors cannot access the meeting room if they have not passed through the lobby

Cu

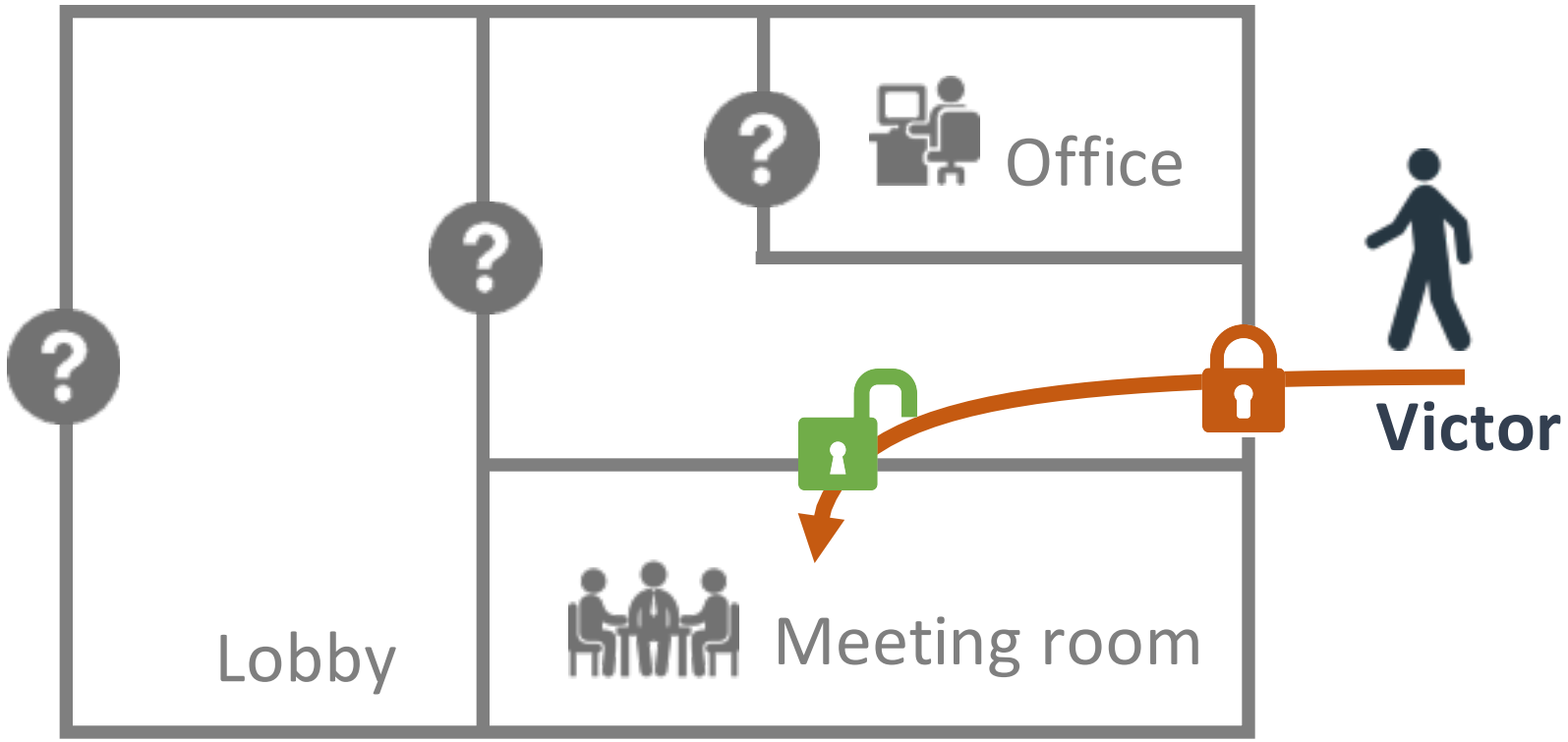
No polici

Phy

Loca
poli

y-one

Example



R1: Visitors can access the meeting room

R2: Visitors cannot access the meeting room if they have not passed through the lobby

Cu

No polici

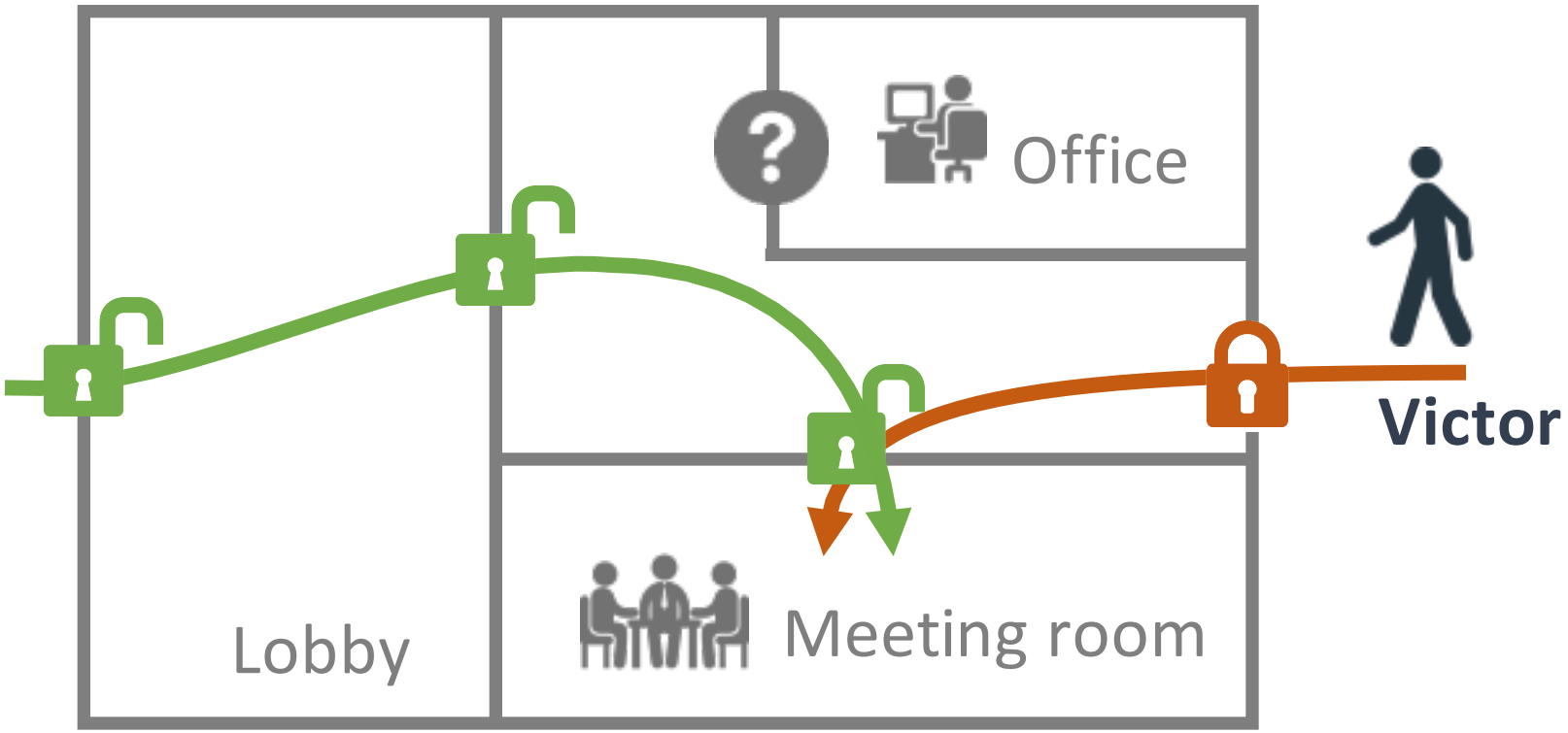
?

Phy

Loca
poli

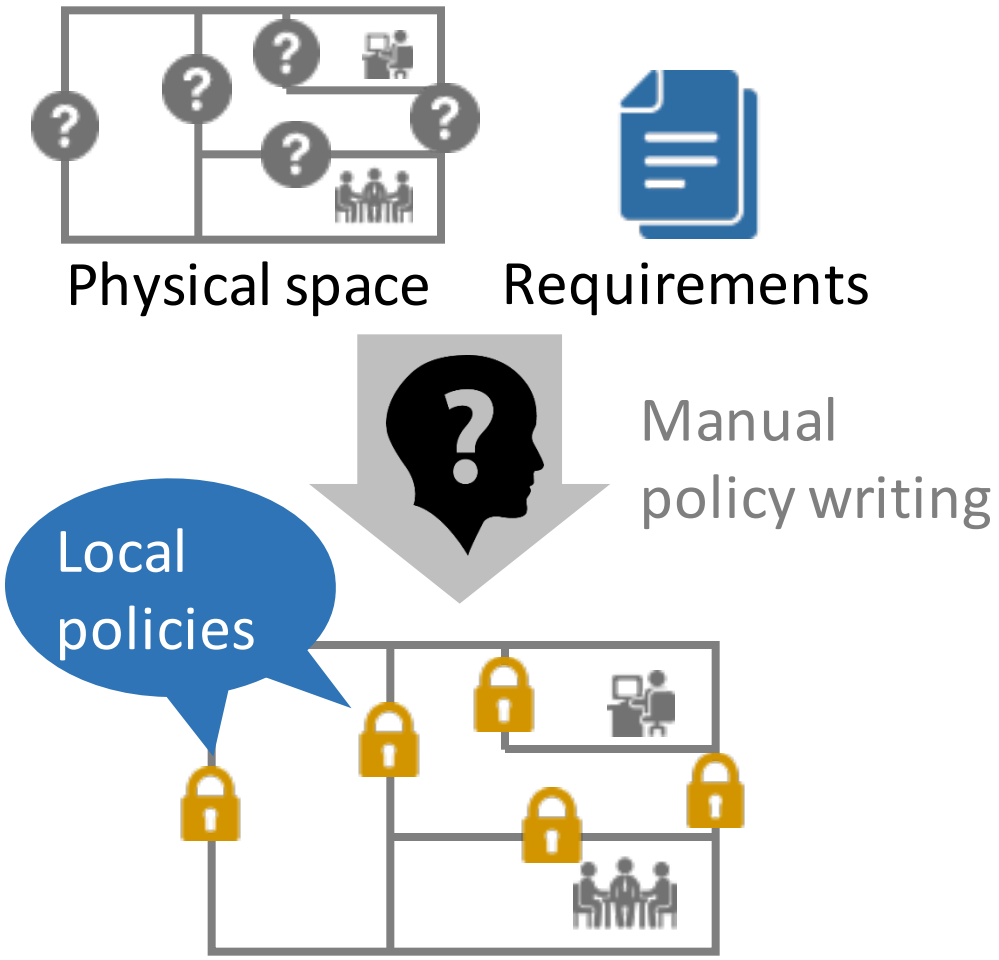
y-one

Example



- ✓ **R1: Visitors can access the meeting room**
- ✓ **R2: Visitors cannot access the meeting room if they have not passed through the lobby**

Current Practice

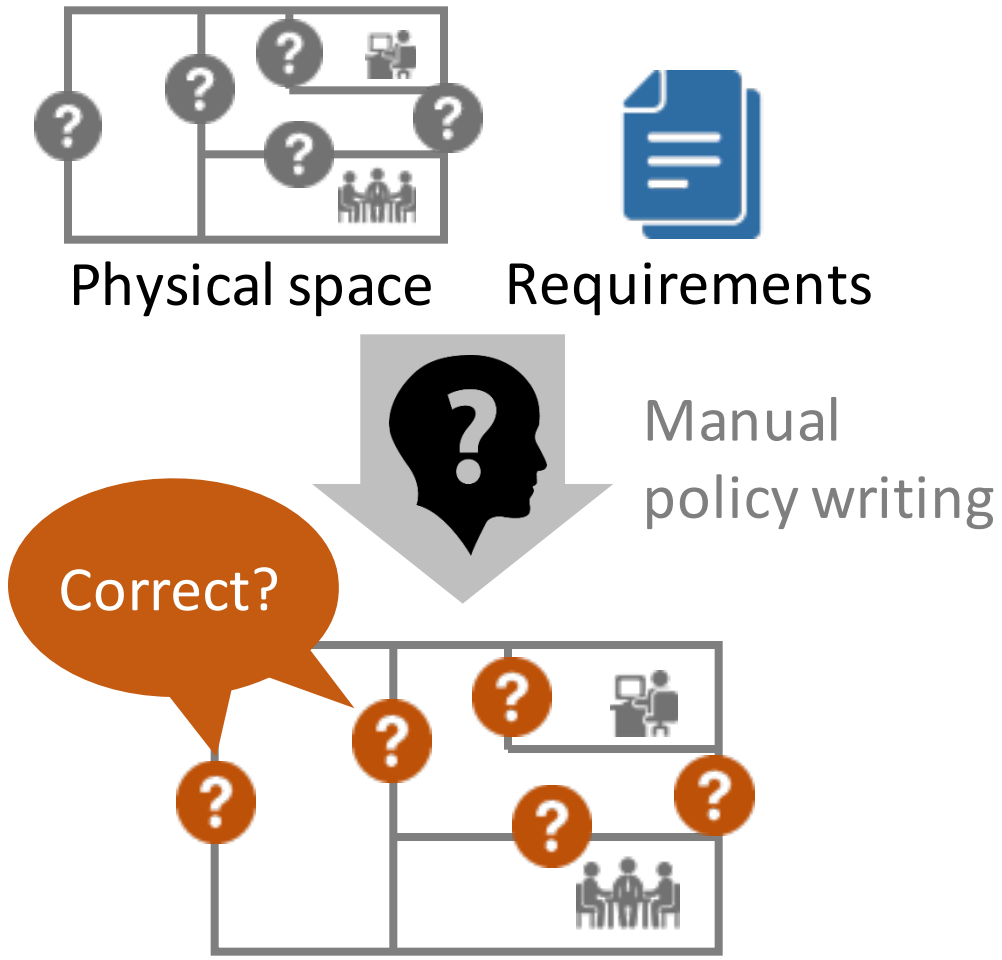


Problems

Cannot satisfy requirements one-by-one

Rewrite policies upon changes to the physical space or requirements

Current Practice



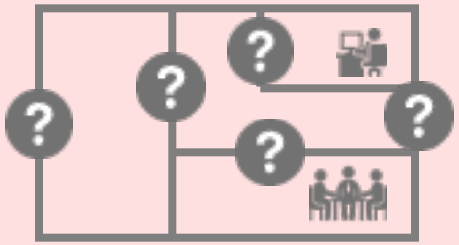
Problems

Cannot satisfy requirements one-by-one

Rewrite policies upon changes to the physical space or requirements

No security guarantees

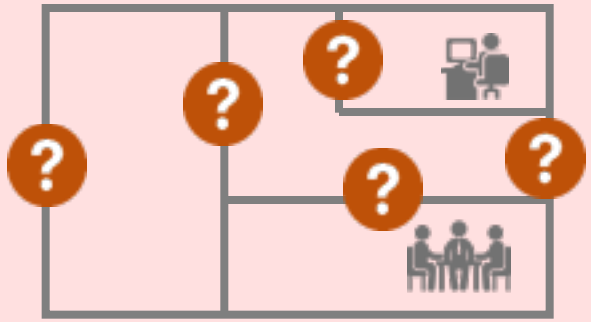
Current Practice



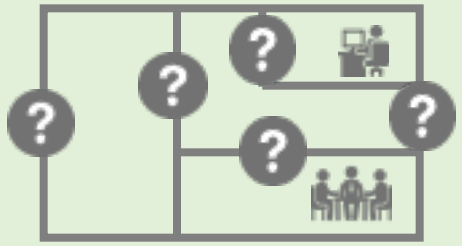
Physical space

Requirements

Manual
policy writing



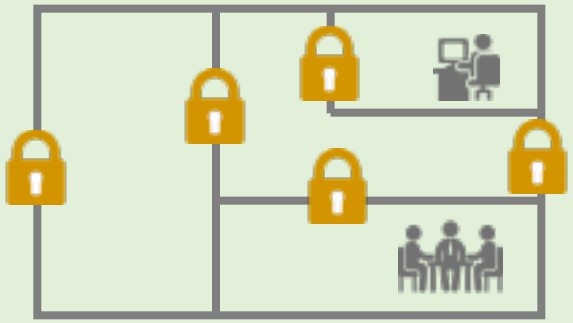
Policy Synthesis



Physical space

Requirements

**Automated
policy synthesis**



Goal

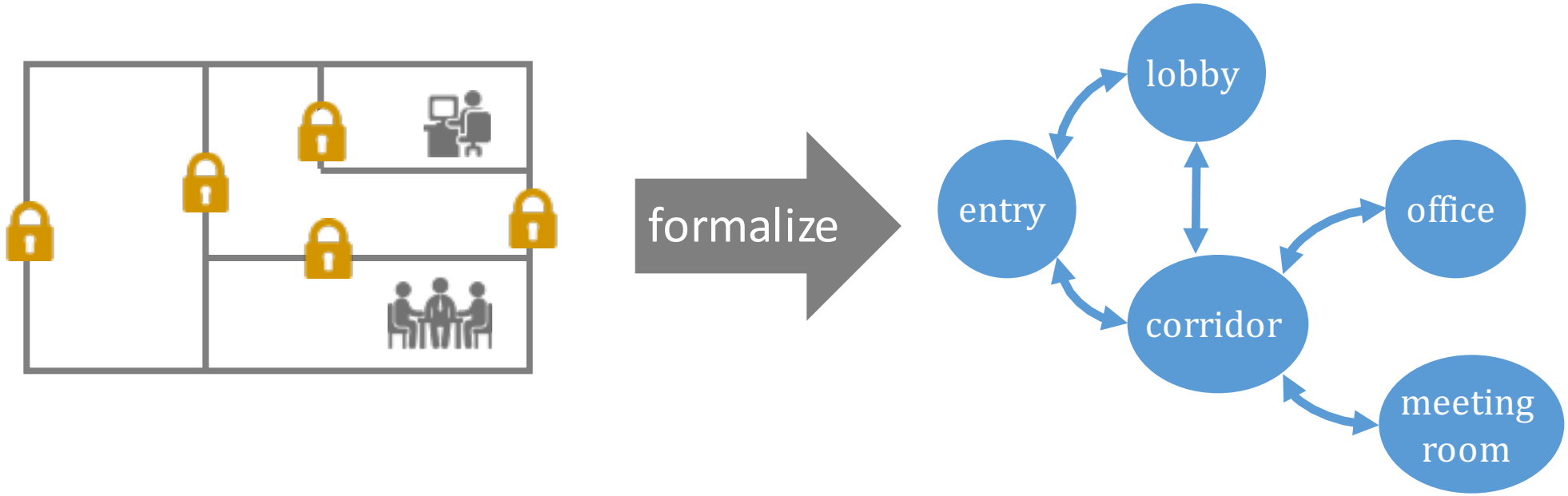
Automatically compute correct local policies for a given physical space and its global requirements

Contributions

- Formalization of physical access control
- **Expressive declarative language** for specifying global requirements
- **Efficient synthesis algorithm** based on SMT solving
- Demonstration of the approach on **realistic case studies**

Formalizing Physical Spaces

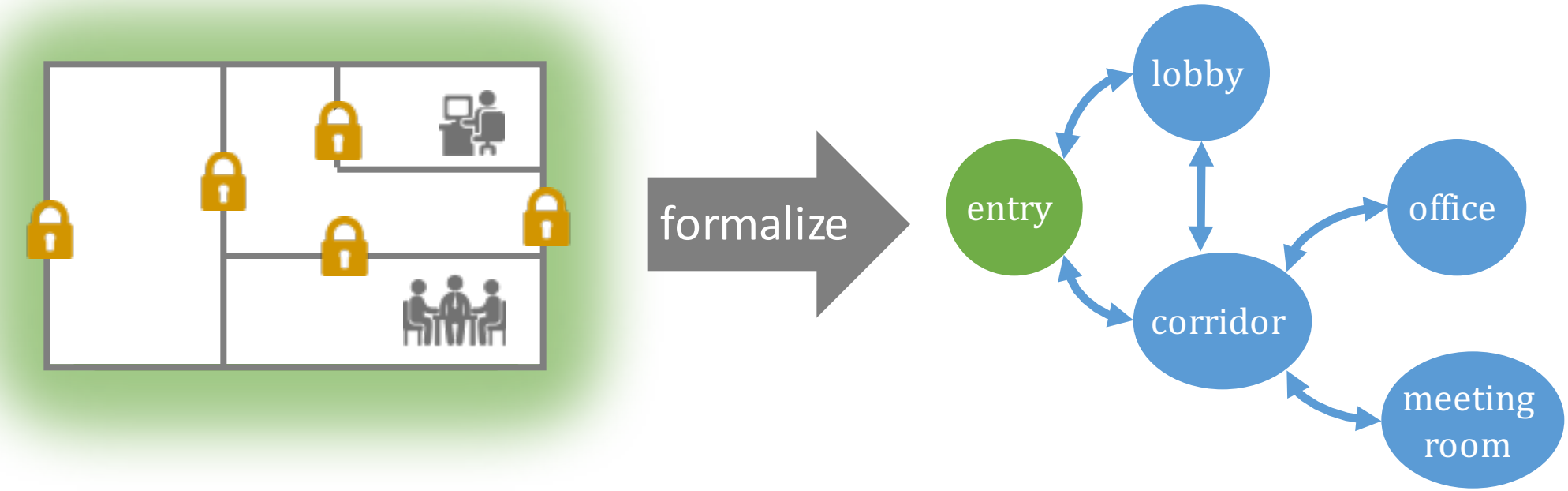
Formalizing Physical Spaces



Enclosed space = Node

Lock = Edge

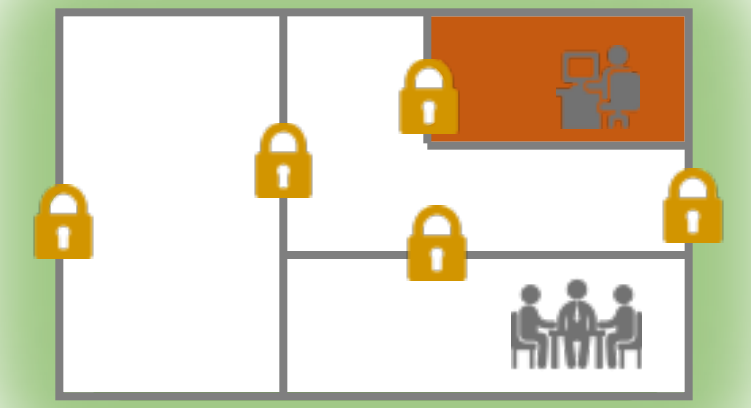
Formalizing Physical Spaces



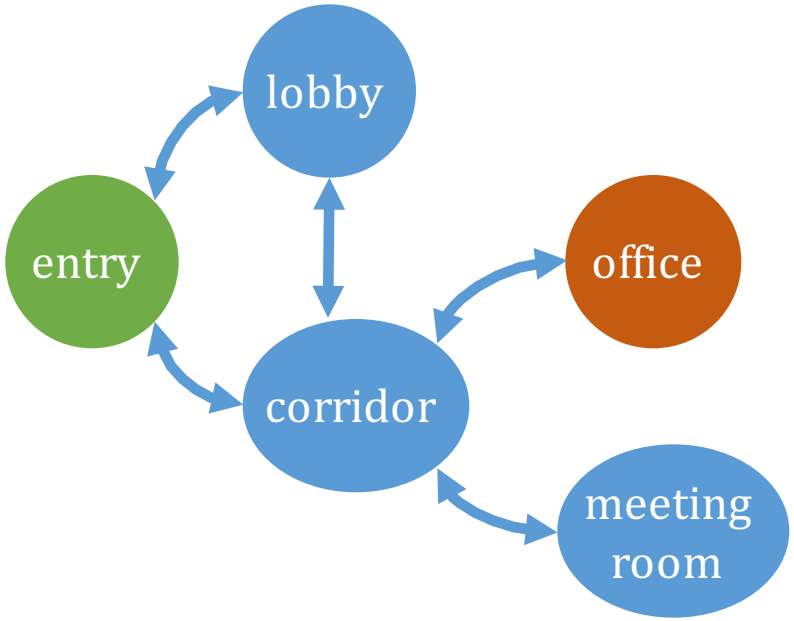
Enclosed space = Node

Lock = Edge

Formalizing Physical Spaces



formalize



Enclosed space = Node

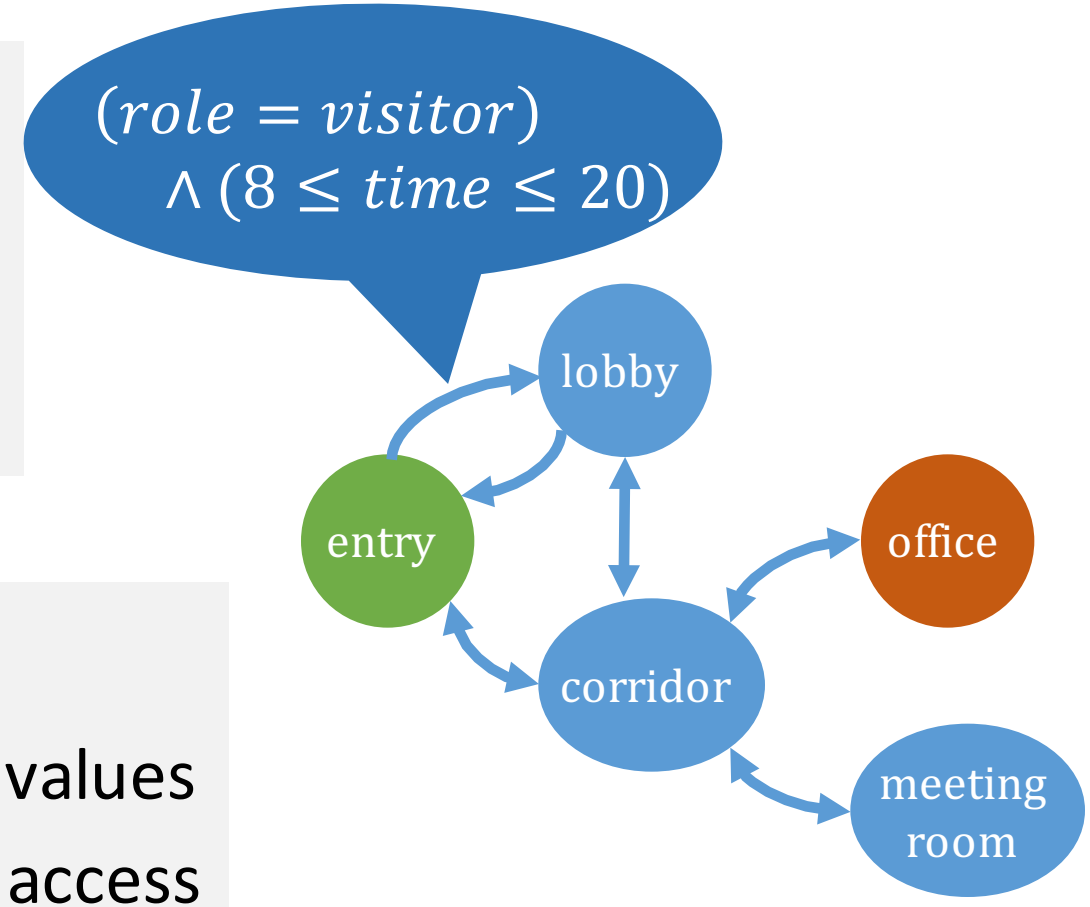
Lock = Edge

Label physical spaces with attributes (e.g., to mark security zones)

Local Policies

Attribute-based policies with:

- Subject attributes (e.g. $roles$)
- Contextual attributes (e.g. $time$)



Local policy semantics

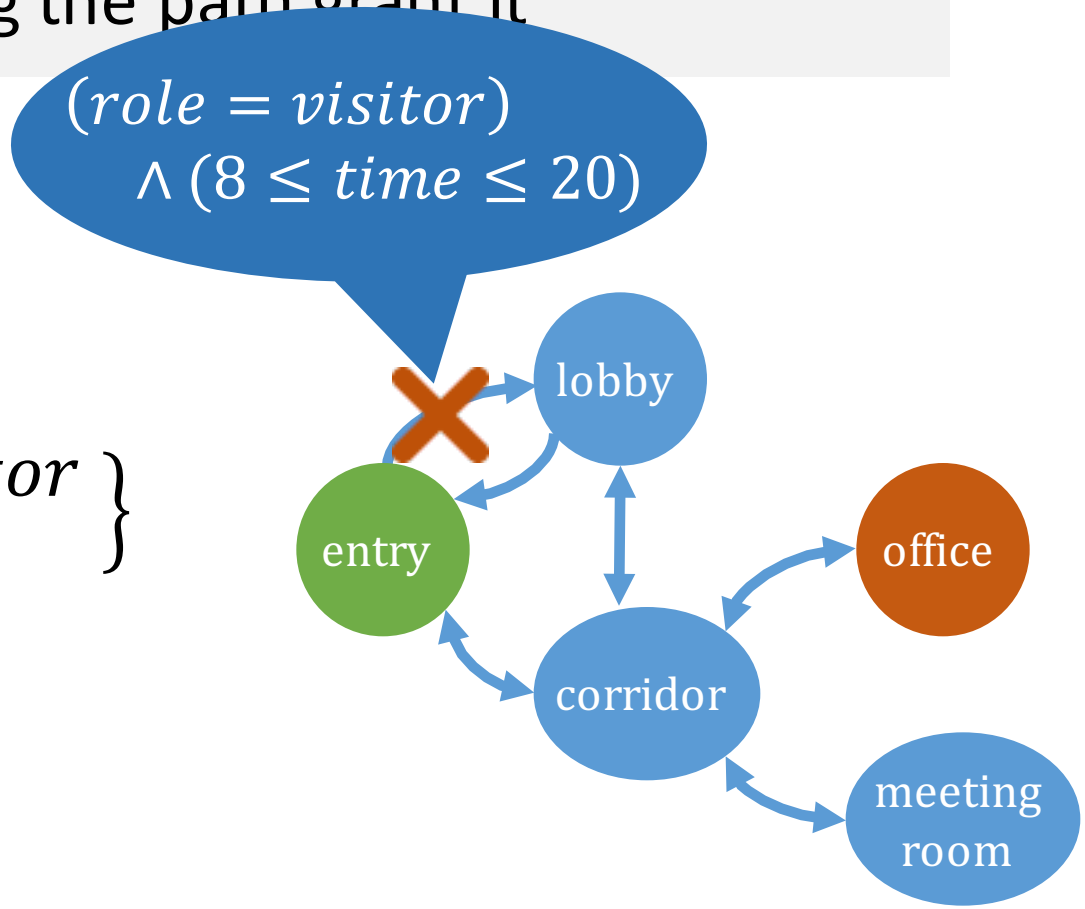
- An access request maps attributes to values
- A lock **grants** an access request if the access request satisfies the lock's local policy

Semantics of Physical Access Control

An access request is authorized along a path if all locks along the path grant it

Example

$$AccReq_1 = \left\{ \begin{array}{ll} role & \mapsto visitor \\ time & \mapsto 6 \end{array} \right\}$$

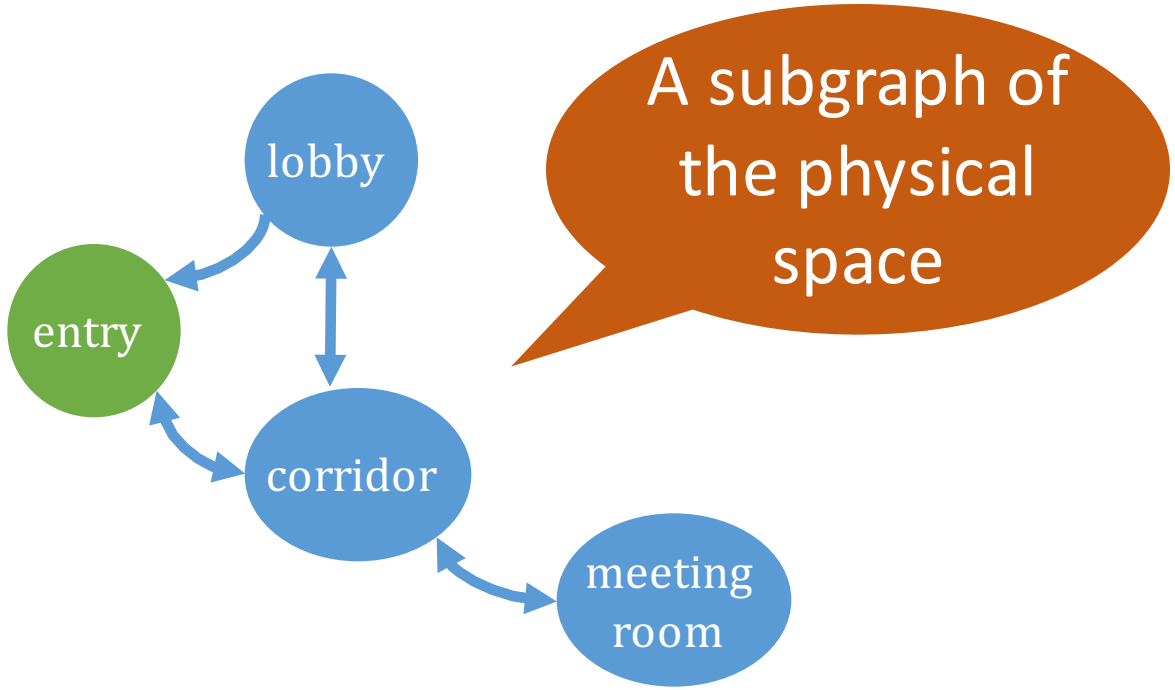


Semantics of Physical Access Control

An access request is authorized along a path if all locks along the path grant it

Example

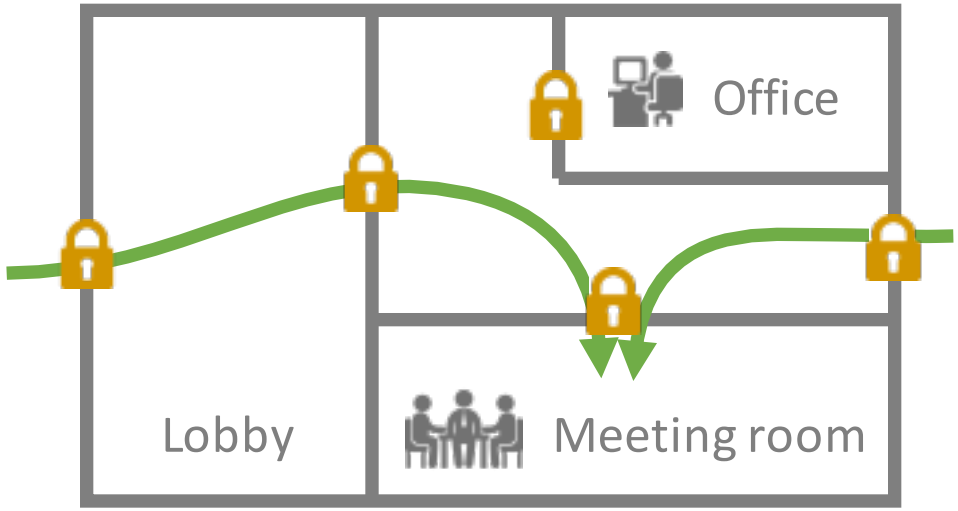
$$AccReq_1 = \left\{ \begin{array}{l} role \mapsto visitor \\ time \mapsto 6 \end{array} \right\}$$



Specifying Global Requirements

Requirement Examples

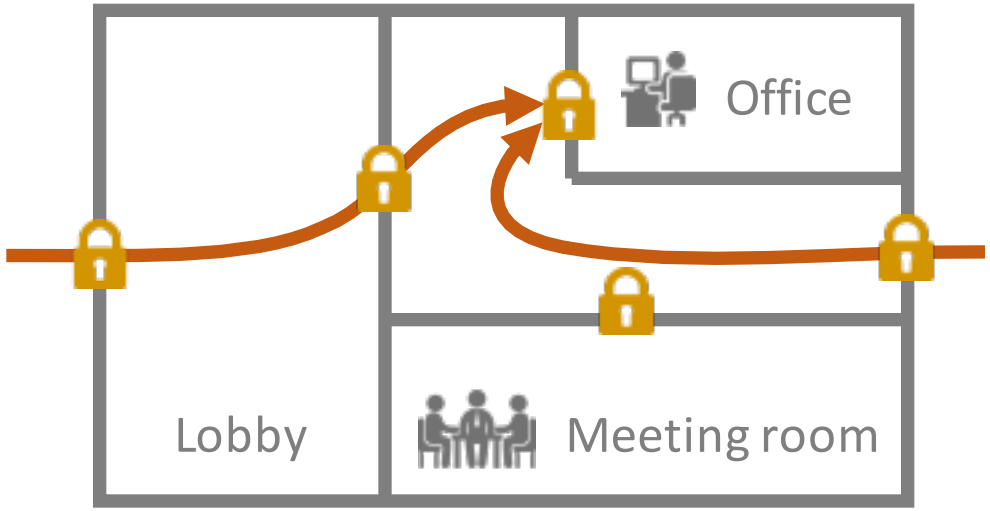
Visitors can access the meeting room



Requirement Examples

Visitors can access the meeting room

Non-employees cannot access the office

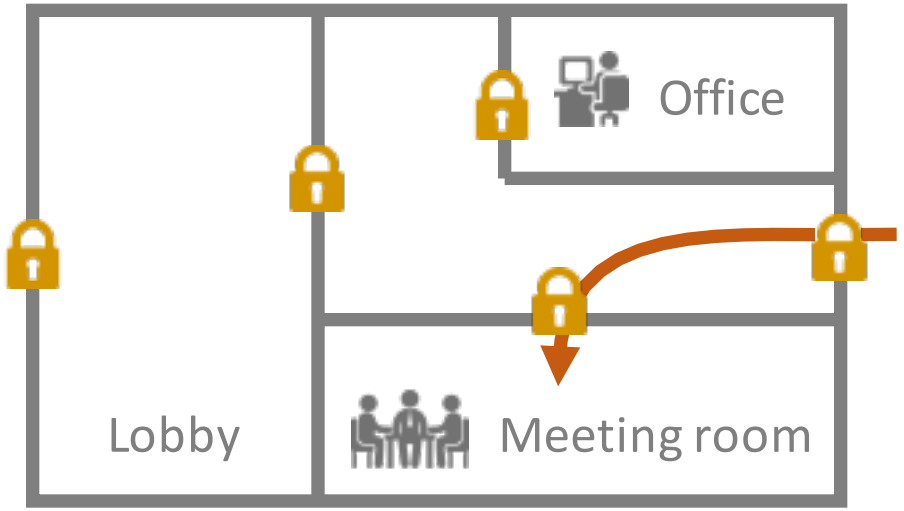


Requirement Examples

Visitors can access the meeting room

Non-employees cannot access the office

Visitors cannot access the meeting room if they have not passed through the lobby



The SpCTL Language

Key features

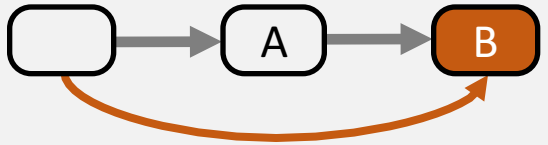
- Subject & contextual attributes
  e.g. $role, time$
- Resource attributes
  e.g. $securityZone$
- Quantification over paths

Common patterns

Permission

Prohibition

Waypointing



Example: $(role = visitor) \wedge (8 \leq time \leq 20) \Rightarrow EF\,(id = mr)$

The SpCTL Language

Key features

Subject & contextual attributes
e.g. $role, time$

Resource attributes
e.g. $securit$

Quant

Constraint over subject & contextual attributes

Common patterns

Permission 

Prohibition 

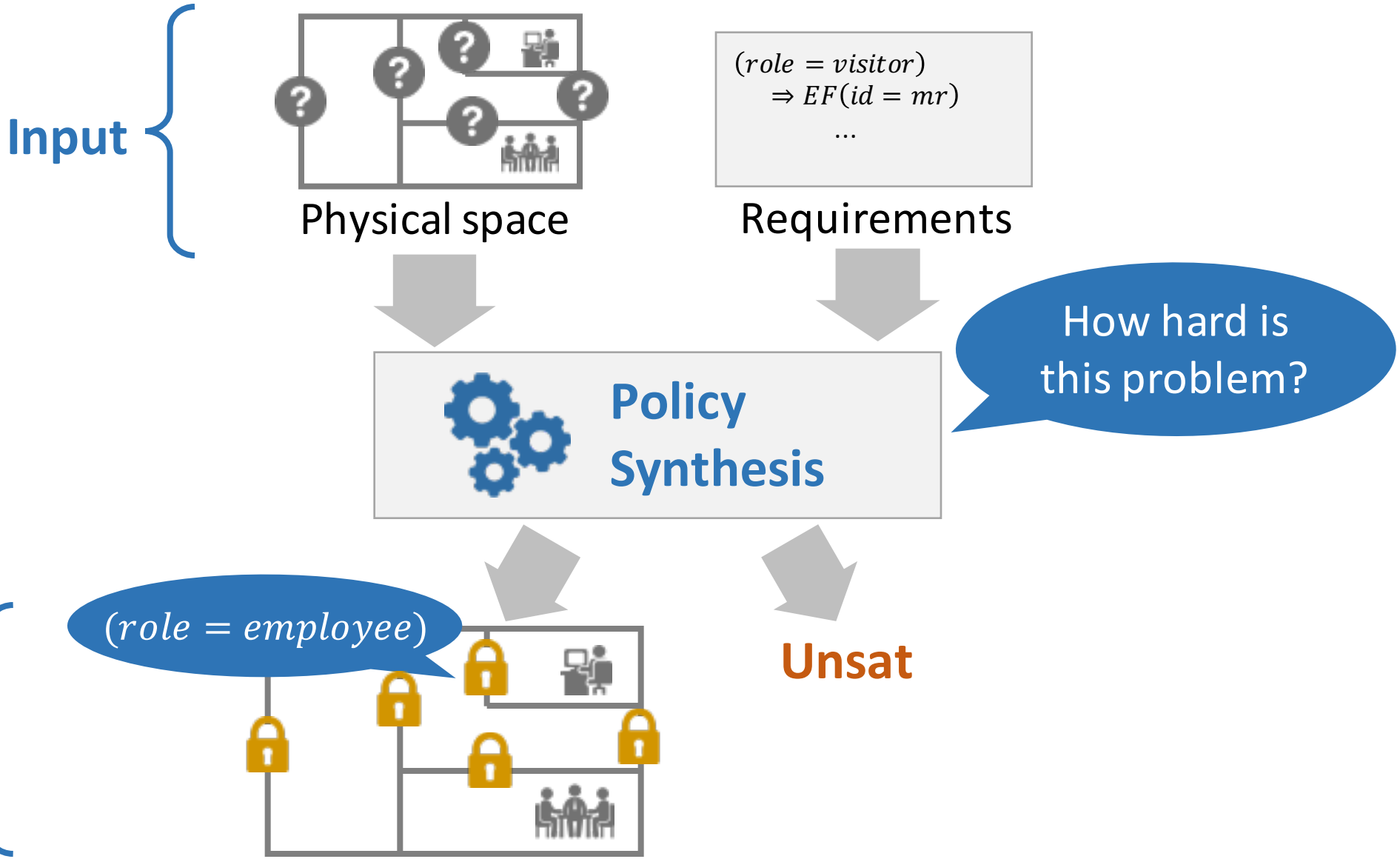
Waypointing 

CTL formula over resource attributes

Example: $(role = visitor) \wedge (8 \leq time \leq 20) \Rightarrow EF\,(id = mr)$

Policy Synthesis Problem

Policy Synthesis Problem



Complexity of Policy Synthesis

Theorem 1. The policy synthesis problem is decidable.

Proof. We give a synthesis algorithm that uses CTL controller synthesis as a subroutine

Theorem 2. The policy synthesis problem is NP-hard.

Proof. Through reduction from propositional satisfiability to policy synthesis

Complexity of Policy Synthesis

Theorem 1. The policy synthesis problem is decidable.

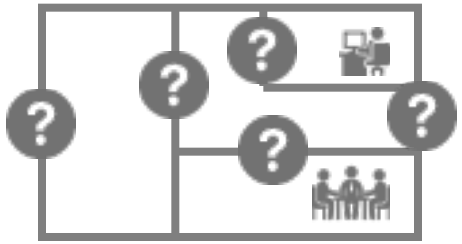
Proof. We give a synthesis algorithm that uses CTL controller synthesis as a subroutine

Unfortunately, the running time of this algorithm is exponential in the number of requirements

Theorem 2. The poli

Proof. Through reduction from propositional satisfiability to policy synthesis

Policy Synthesis using SMT Solving



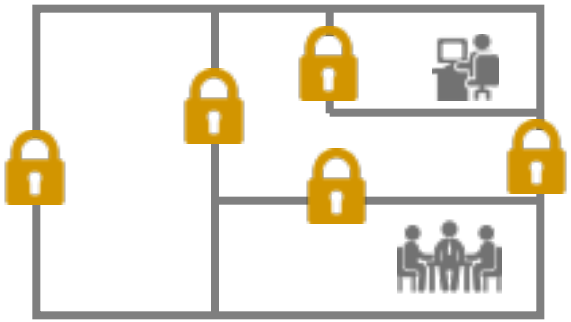
Physical space

$(role = visitor)$
$\Rightarrow EF(id = mr)$
$\ldots$

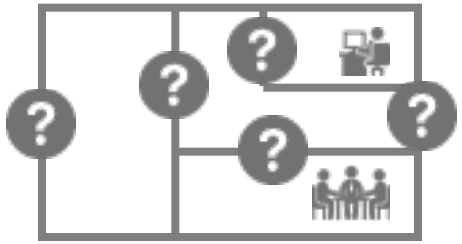
Requirements

$\exists\forall$ SMT Solving

Unsat



Policy Synthesis using SMT Solving



Physical space

$(role = visitor)$
$\Rightarrow EF(id = mr)$
$\ldots$

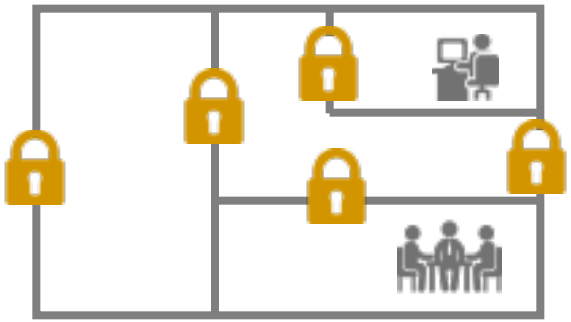
Requirements

$\exists\forall$ SMT Solving

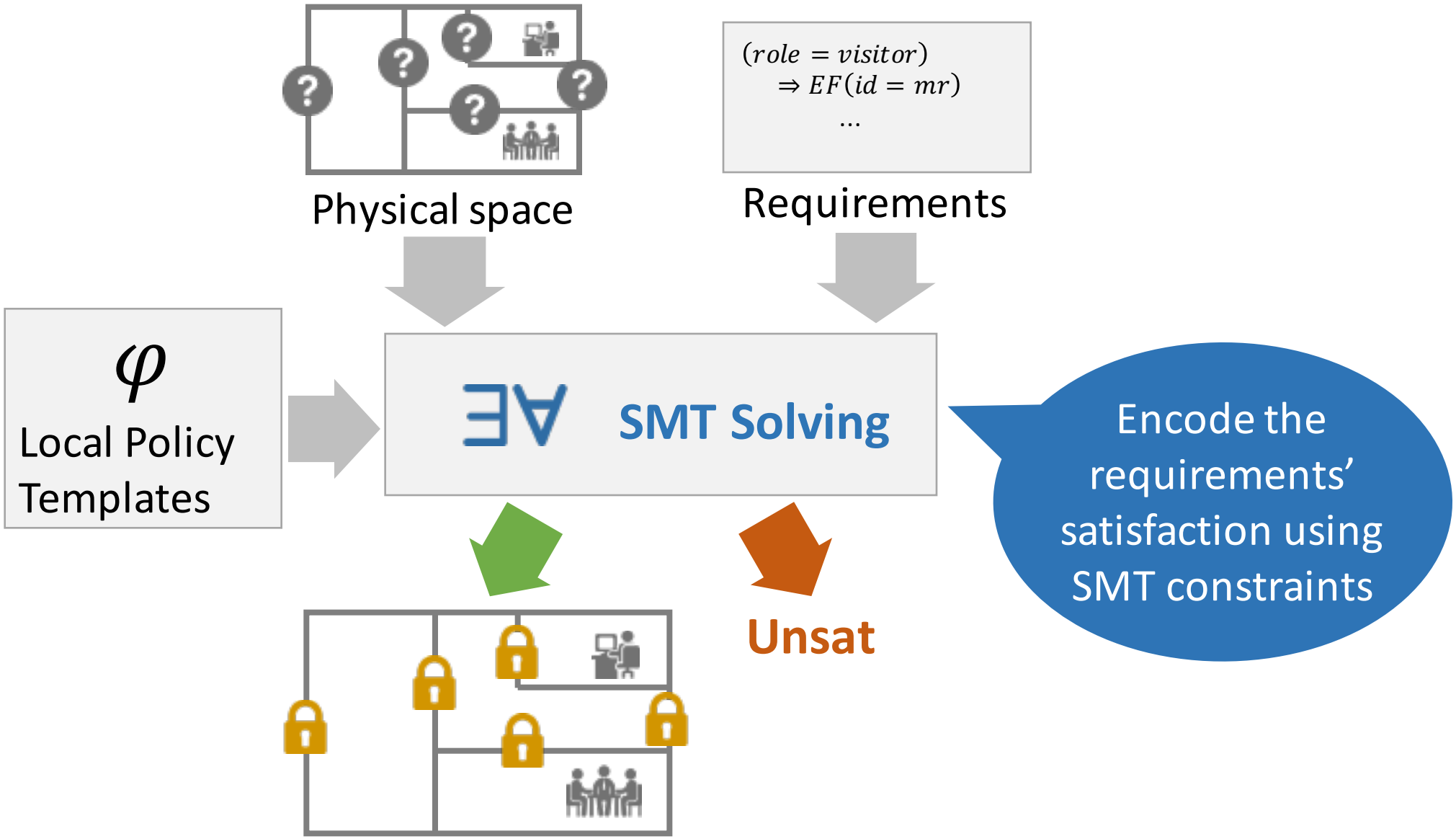
A model identifies correct local policies

Unsat

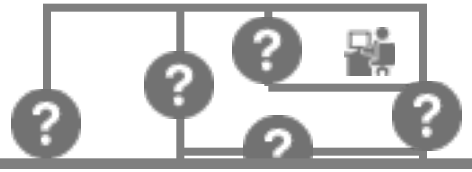
Encode the requirements' satisfaction using SMT constraints



Policy Synthesis Algorithm



Policy Synthesis Algorithm



$(role = visitor)$
$\Rightarrow EF(id = mr)$

Example Template

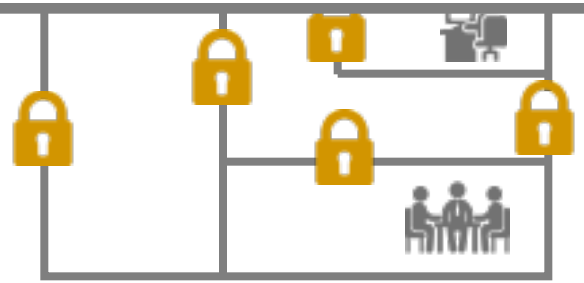
$(? = ?) \wedge (? \leq ? \leq ?)$

(example instantiation)

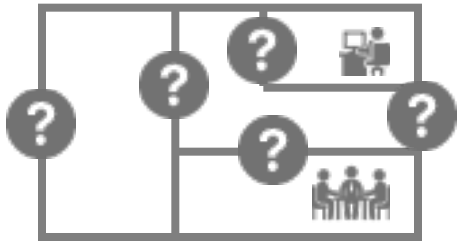
$(role = visitor) \wedge (8 \leq time \leq 20)$

s'
ing
nts

Unsat



Policy Synthesis Algorithm



Physical space

$(role = visitor)$
$\Rightarrow EF(id = mr)$
$\ldots$

Requirements

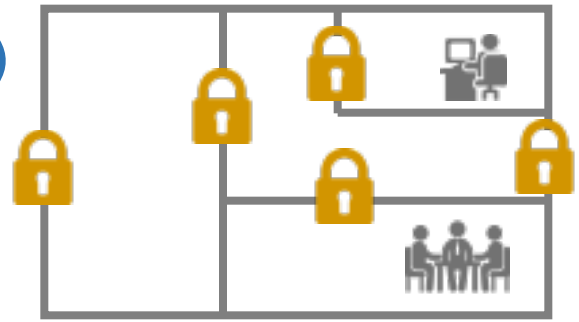
Reduce search space

φ
Local Policy Templates

$\exists\forall$ SMT Solving

Encode the requirements' satisfaction using SMT constraints

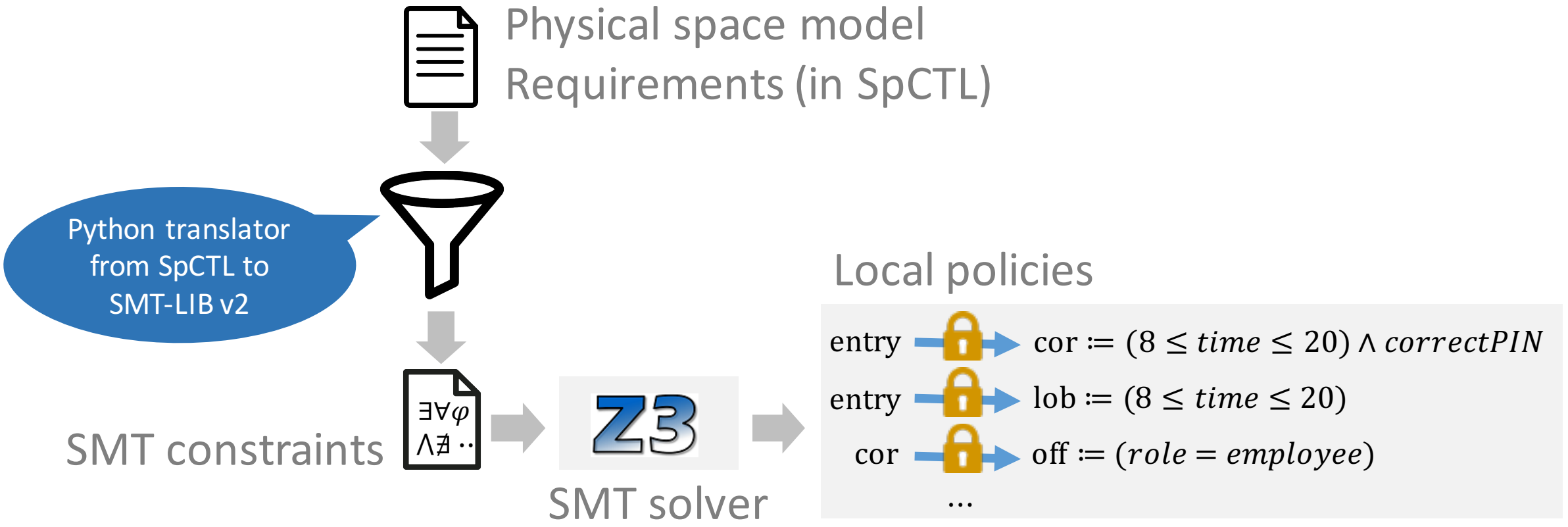
Synthesize concise local policies



Unsat

Implementation and Evaluation

Implementation



Our system is publicly available

https://github.com/ptsankov/SpCTL

Evaluation: Case Studies

KABA Headquarters

Physical spaces	20
Locks	41
Requirements	10
Synthesis time	**25s**

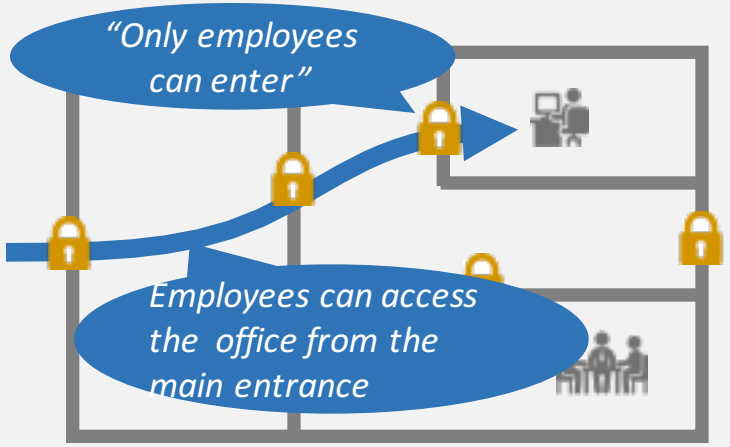
ETH's CS Department

Physical spaces	66
Locks	127
Requirements	14
Synthesis time	**10s**

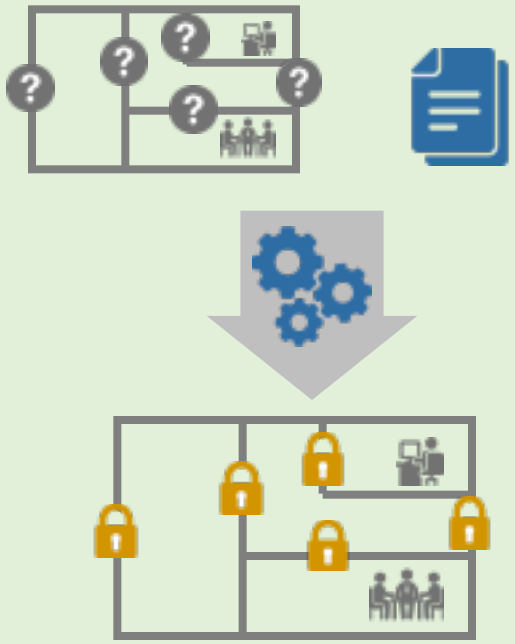
Airport Terminal

Physical spaces	13
Locks	32
Requirements	15
Synthesis time	**2s**

Summary



Global requirements vs local enforcement



Policy synthesis framework

Approach scales to realistic problems