Modelling interfaces in distributed systems: some first steps

David Pym

UCL and Alan Turing Institute

London

Modelling distributed systems: basic concepts

- Basic concepts of distributed systems
  - Location: the basic architecture
  - Resource: created, consumed, moved by the processes
  - Process: the services that the system provides
- Situated in
  - Environment: *structure* not modelled, just events
- These may be composed partially of other models of interest, so need composition
- Mathematically, seek to employ minimal viable structure
- Concerned here with *practical modelling*, with motivations from *security policy*

Modelling distributed systems: basic mathematical set-up

- Location
  - Topological structure: e.g., directed graphs
- Resource
  - Combinatorial structure: e.g., partial monoids, possibly ordered (cf. the logic BI's resource semantics, which gives rise to Separation Logic)
- Process
  - Synchronous structure (for modelling purposes): e.g., SCCS + integration with resources
- Environment
  - Stochastic representation: events are incident upon a model system from outside

Modelling distributed systems: basic mathematical set-up

- Basic operational judgement:

$$L, R, E \xrightarrow{a} L', R', E'$$

- Some rules (omitting locations for brevity):

$$\frac{\mu(a, R) = R'}{R, a : E \xrightarrow{a} R', E} \qquad \frac{R, E \xrightarrow{a} R', E' \quad S, F \xrightarrow{b} S', F'}{R \otimes S, E \times F \xrightarrow{ab} R' \otimes S', E' \times F'}$$

$$\frac{R_i, E_i \xrightarrow{a} R_i', E_i'}{R_1 \oplus R_2, E_1 + E_2 \xrightarrow{a} R_i', E_i'} \quad i = 1, 2$$

- A bunch of laws for μ, $\otimes$, and $\oplus$
- Resource-process equivalence is bisimulation, written ~
- Cf. Concurrent Separation Logic

A (bunched) modal logic

$$
\begin{array}{lll}
\phi & ::= & \mathrm{p} \mid \bot \mid \top \mid \phi \vee \phi \mid \phi \wedge \phi \mid \phi \rightarrow \phi \\
 & & \mid \langle a \rangle \phi \mid [a]\phi \\
 & & \mid I \mid \boxed{\phi * \phi} \mid \phi \mathrel{-\!\!*} \phi \\
 & & \mid \langle a \rangle_\nu \phi \mid [a]_\nu \phi
\end{array}
$$

In a given model, a truth-functional judgement: $R, E \models \phi$

$$R, E \models \phi_1 \wedge \phi_2 \text{ iff } R, E \models \phi_1 \text{ and } R, E \models \phi_2$$

$$R, E \models \langle a \rangle \phi \text{ iff for some } R, E \xrightarrow{a} R', E',\ R', E' \models \phi$$

$$
\boxed{
\begin{array}{ll}
R, E \models \phi_1 * \phi_2 \text{ iff} & \text{for some } R_1 \otimes R_2 = R \text{ and } E_1 \times E_2 \sim E, \\
 & R_1, E_1 \models \phi_1 \text{ and } R_2, E_2 \models \phi_2
\end{array}
}
$$

$$
\begin{array}{ll}
R, E \models \langle a \rangle_\nu \phi \text{ iff} & \text{for some } S,\ S' \text{ s.t. } R \otimes S, E \xrightarrow{a} R' \otimes S', E', \\
 & R' \otimes S', E' \models \phi
\end{array}
$$

Other similar things, some choices for the last one

Basic meta-theory

- Logical (declarative) equivalence:

$$R_1, E_1 \equiv R_2, E_2 \text{ iff for all } \phi,\ R_1, E_1 \models \phi \text{ iff } R_2, E_2 \models \phi$$

- Bisimulation (operational) equivalence:

$$R_1, E_1 \sim R_2, E_2$$

- Soundness and completeness (Hennessy-Milner-van Bentham equivalence):

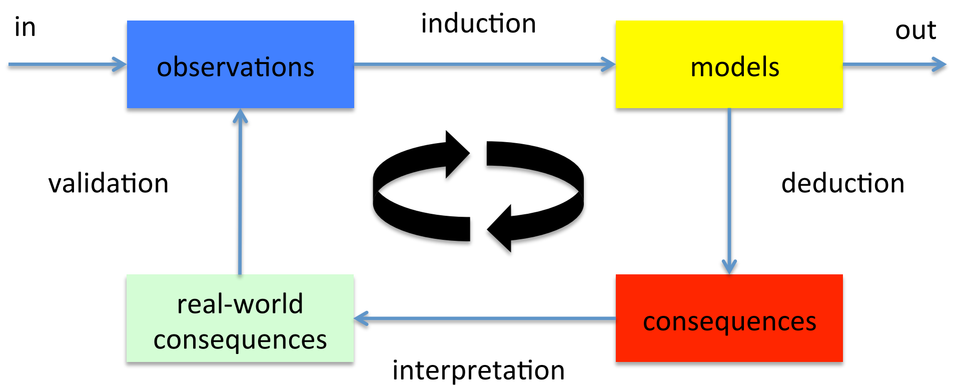
$$\text{for all } R_1, E_1, \quad R_1, E_1 \sim R_2, E_2 \text{ iff } R_1, E_1 \equiv R_2, E_2$$

Basic meta-theory

- Hennessy-Milner completeness is not as straightforward as might perhaps be imagined
- In basic resource semantics, based on ordered monoids of resource elements, it holds only for fragments of the modal logic
- Multiplicative implication and multiplicative modalities problematic
- Need the combinatorial structure of $\oplus$ and $\otimes$ to track evolutions of + and x
- Several papers (MSCS, TCS, JLC, others): http://www.cs.ucl.ac.uk/staff/D.Pym/recent.htm

Building models

- Classical mathematical modelling approach using these tools

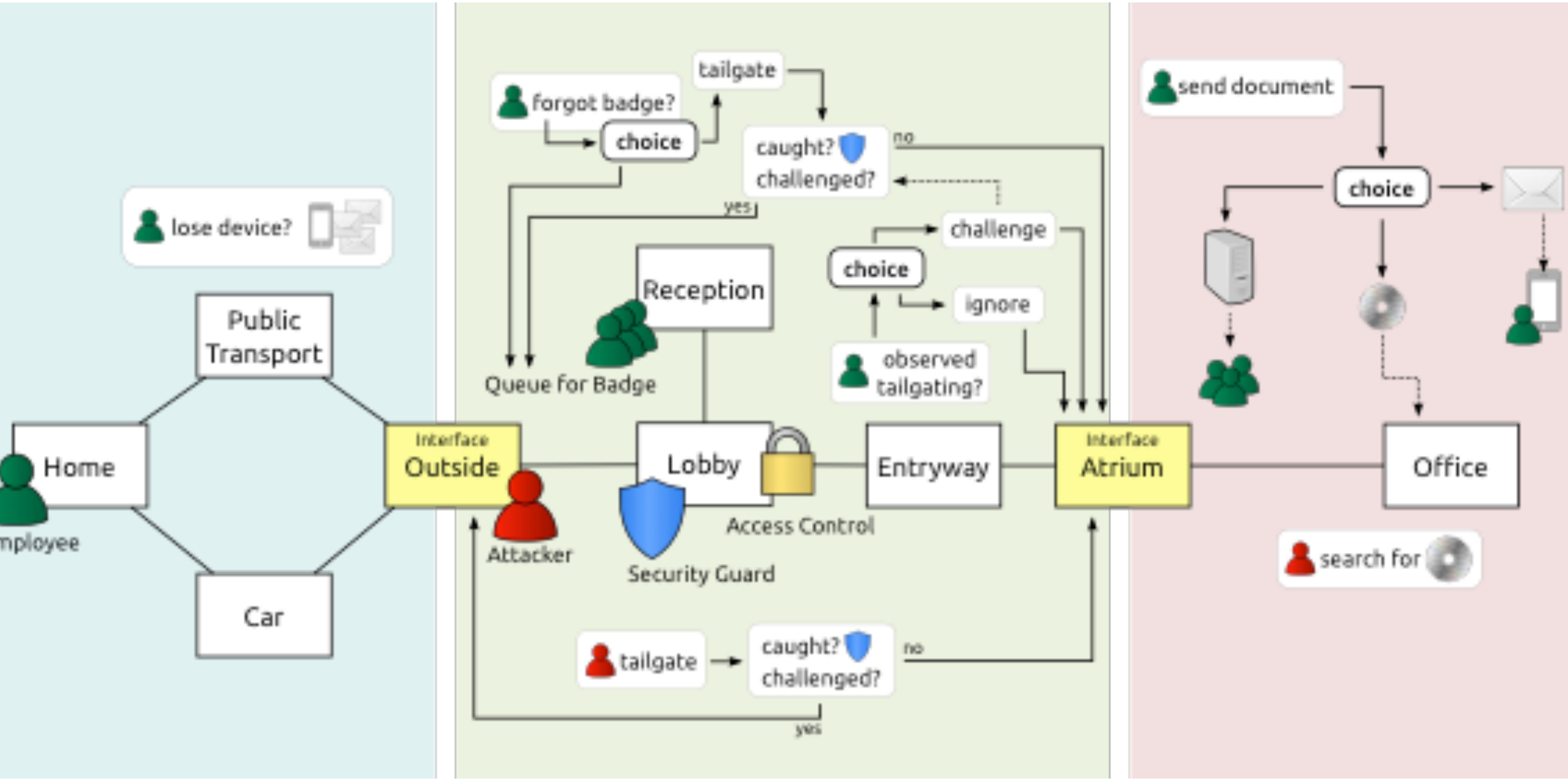


- Early versions deployed with Hewlett-Packard and its customers, and more recently in projects in the GCHQ RISCS
- Currently aiming for policy modelling apps in the Turing Institute; lots of big industry partners
- Several papers at http://www.cs.ucl.ac.uk/staff/D.Pym/recent.htm
- julia code at https://github.com/tristanc/SysModels

Aside: building models

- Approach is essentially scale-free
- Abstraction level therefore chosen to fit problem
- Predictions explored using simulations
- Model checking also possible (though much less developed at this point)
- The map is not the territory (Alfred Korzybski)
- Time-value of models

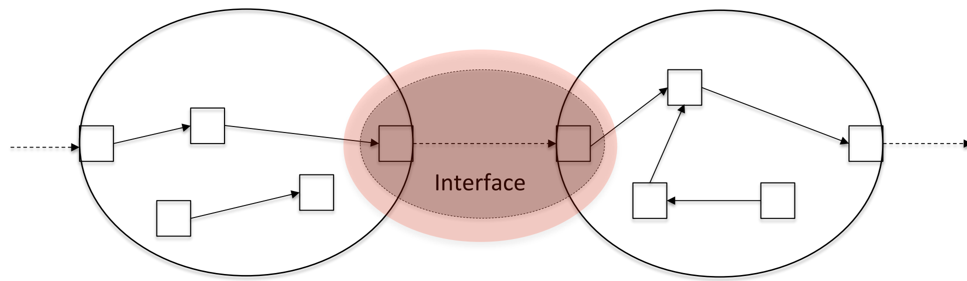
Example: security modelling



Interfaces: basic concepts

- Mediate composition of models
- Build on the structure of distributed systems models, quite pragmatically
- In practice, must reflect
  - the locations involved,
  - the resources involved, and
  - processes/actions crossing the boundaries
- Note that models are being substituted for environment

Interfaces: sketch of basic mathematical set-up



- Implement the distributed systems model:
  - Location graph labelled with resources
  - Explicitly identify actions with associated locations in interfaces
- Each model comes with a specified set of interfaces, specifying input/output locations, with associated actions
- Decent basic algebraic properties: commutative, associative composition of models with compatible interfaces

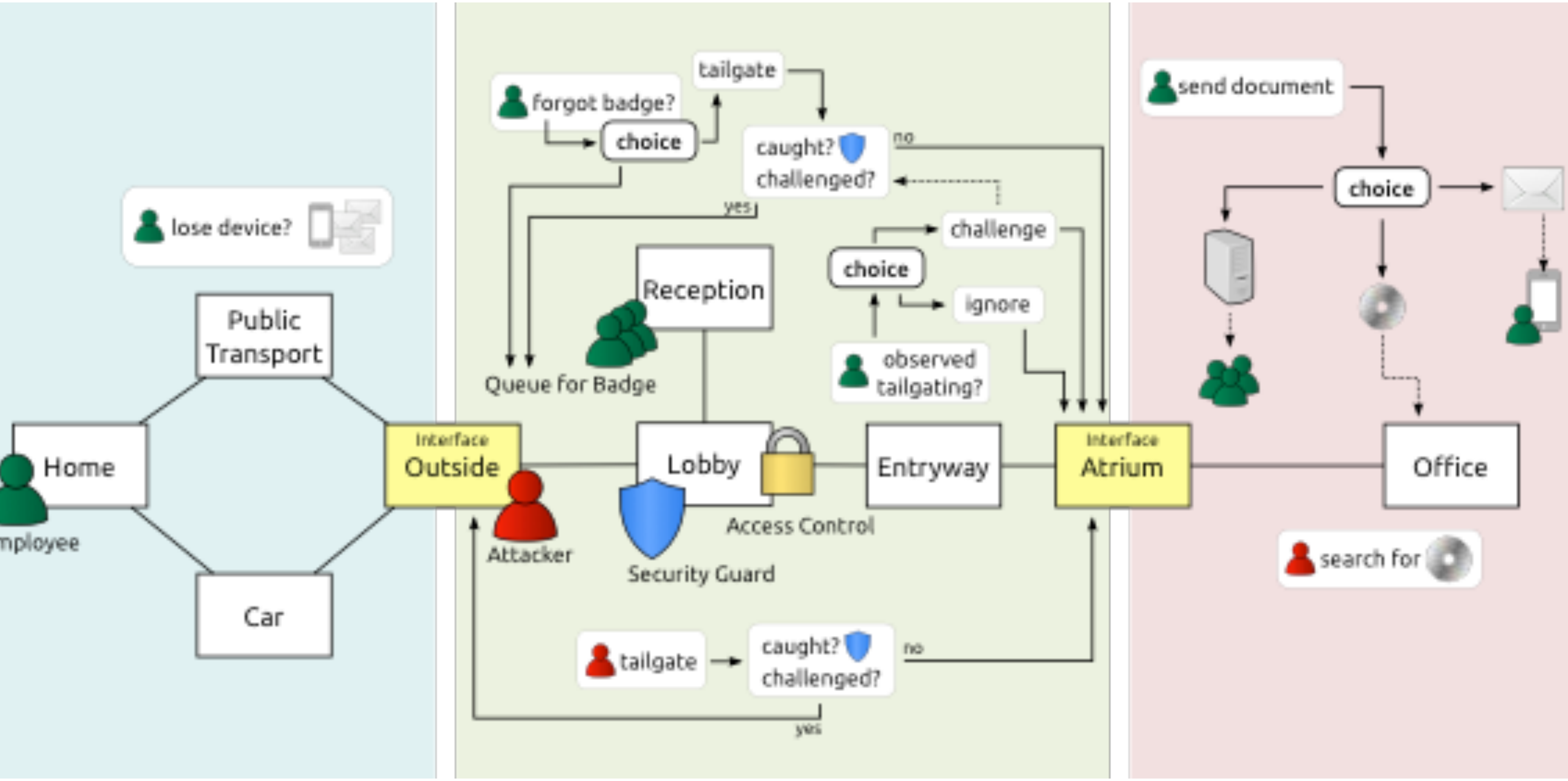
Interfaces: sketch of basic mathematical set-up

- Implement models as tuples

$$M = (\mathcal{G}(\mathcal{V}[R], \mathcal{E}), \mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{I})$$

- Here
  - Graph with resource-labelled vertices
  - Sets of actions, processes, and *located actions*
  - A set $\mathcal{I}$ of *interfaces*
- An interface $I \in \mathcal{I}$ on a model is a tuple of (disjoint) input and output locations and located actions (In, Out, L)

Example: security modelling



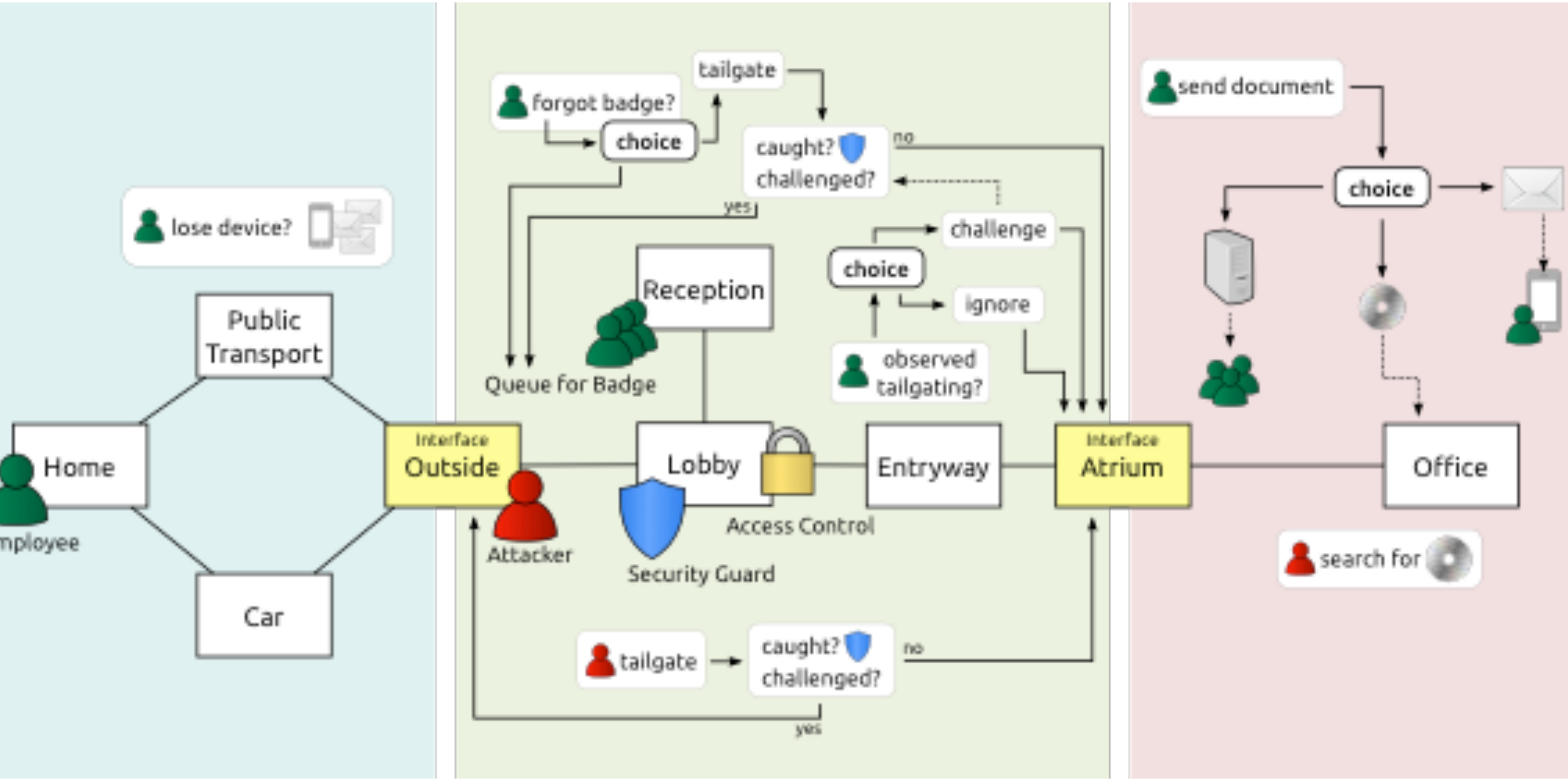
Interfaces: the frame property

- Supports compositional reasoning: $M_1 \;{}_{I_1}|_{I_2}\; M_2$
- The Frame Rule (think of Hoare's program logic and CSL):

$$\frac{\{\phi\}\ (M \xrightarrow{a} M')\ \{\psi\}}{\{\phi * \chi\}\ (M \mid N \xrightarrow{a} M' \mid N)\ \{\psi * \chi\}} \quad N \models \chi, \text{ where } N \not\xrightarrow{a}$$

- Side-condition restricts evolution to part of model not in the interface
- Correctness reasoning can then be restricted to the *interfaces themselves*
- This gives *local* reasoning about models in their *global* context; that is, compositionality

Example: security modelling



Next steps

- Refine definition of interface, useful abstractions
- Some underpinning logical theory
- The Frame Rule in theory and practice; cf. (Concurrent) Separation Logic's theory and implementation of local reasoning: *abduction* important here?
- Applications to big-scale systems
  - Networking
  - Distributed databases and their consistency
  - Supply chains
- Deliver tools for reasoning about big-scale systems
- Small-scale systems: weak memory

Thank you

Modelling distributed systems: basic mathematical set-up

- Other key combinators
  - Hiding

$$\frac{R \circ S, E \xrightarrow{a} R' \circ S', E'}{R, \nu S.E \xrightarrow{\nu S.a} R', \nu S'.E'} \quad \mu(\nu S.a, R) = R'$$

  - Generalizes restriction (build a term model for resources; partial monoid of actions)
- Sequential composition
- Fixed points

A (bunched) modal logic

- Other logical operators
  - Additive and multiplicative quantifiers (over actions)

$$R, E \models \exists_\nu x.\phi \quad \text{iff} \quad \text{there exist } S,\ F, \text{ and } a \text{ s.t. } R, E \sim R, \nu S.F$$
$$\text{and } R \circ S, F \models \phi[a/x]$$

- Systematic logical treatment in recent joint work with Galmiche, Courtault, and Kimmel
- Applications in access control
  - Roles: $E \propto F$
  - Corresponding (via simulation) 'says' modality: $\{E\}\phi$

References

- G. Anderson and D. Pym. A Calculus and Logic of Bunched Resources and Processes. *Theoretical Computer Science* 614:63-96, 2016.
- D. Galmiche, J.-R. Courtault, D. Pym. A Logic of Separating Modalities. *Theoretical Computer Science* 637, 30-58, 2016.
- M. Collinson and D. Pym. Algebra and Logic for Resource-based Systems Modelling. *Mathematical Structures in Computer Science* 19:959-1027, 2009. doi:10.1017/S0960129509990077.
- M. Collinson, B. Monahan, D. Pym. *A Discipline of Mathematical Systems Modelling*. College Publications, 2012.

More references

- T. Caulfield and D. Pym. Modelling and Simulating Systems Security Policy. *Proc. SIMUTools 2015*, ACM Digital Library, SIMUtools 2015. doi: 10.4108/eai.24-8-2015.2260765.
- T. Caulfield and D. Pym. Improving Security Policy Decisions with Models. *IEEE Security and Privacy*, 13(5), 34-41, September/October 2015.

- The julia package used for creating system models may be obtained from GitHub: https://github.com/tristanc/SysModels