Mobility

Luca Cardelli

Microsoft Research

Agay, March 2002

Reflecting joint work with Luís Caires, Andrew D. Gordon.

Global Communication

Isn't π-calculus good(/bad) enough?

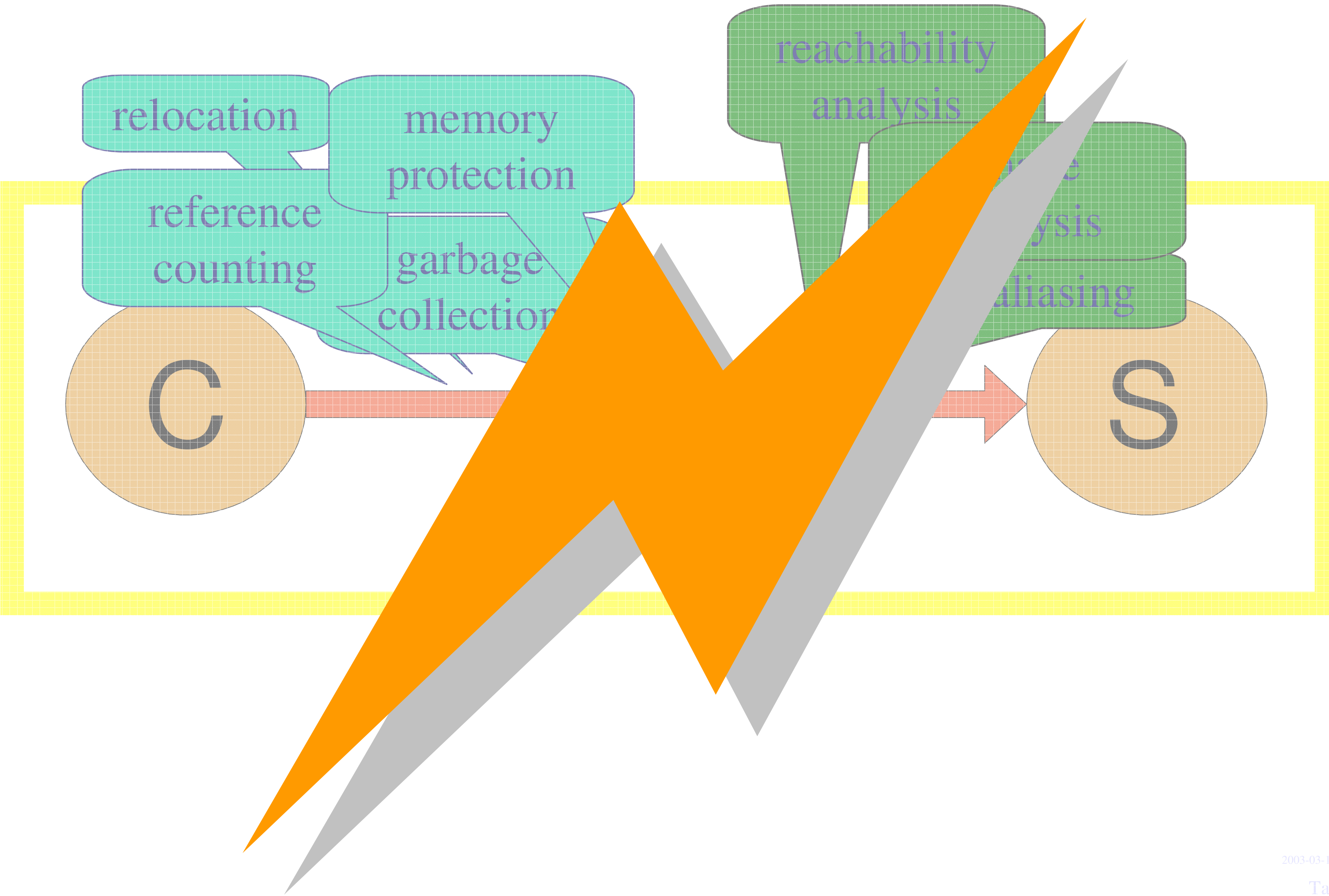
- Most process calculi use a powerful *channel* abstraction.
- This is "too abstract" for global communication:
  failure modes get increasingly harder to ignore.
- Channels abstract *wires*.
  What kind of wires do we actually need to model?

Two "Paradoxes" of global communication:

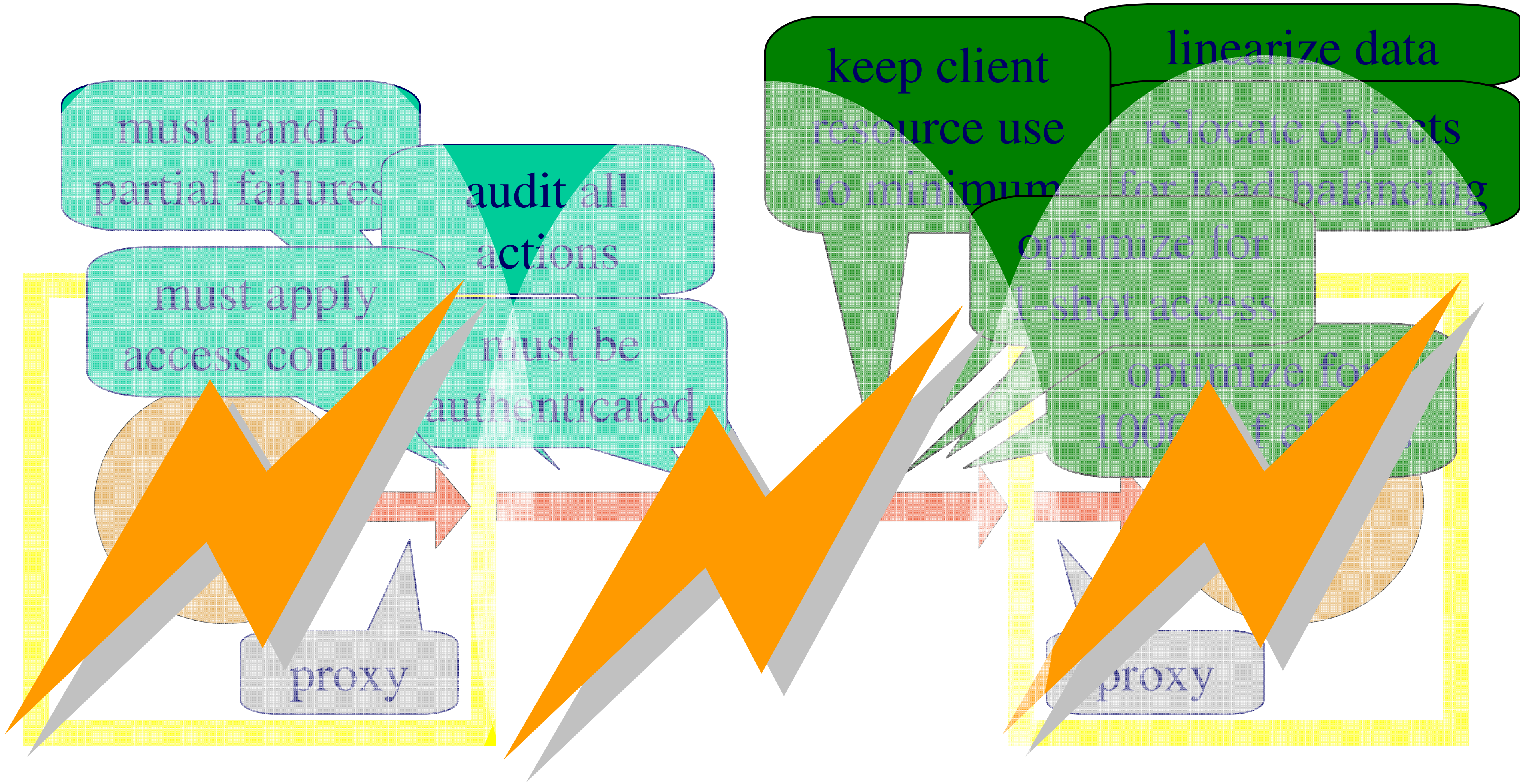
- Wires are very, very complicated.
  Most of Computer Science is about modeling or implementing wires.
- Even when nothing goes wrong, still things don't work.
  Global Murphy's Law.

Ditch channels, but keep π-names.

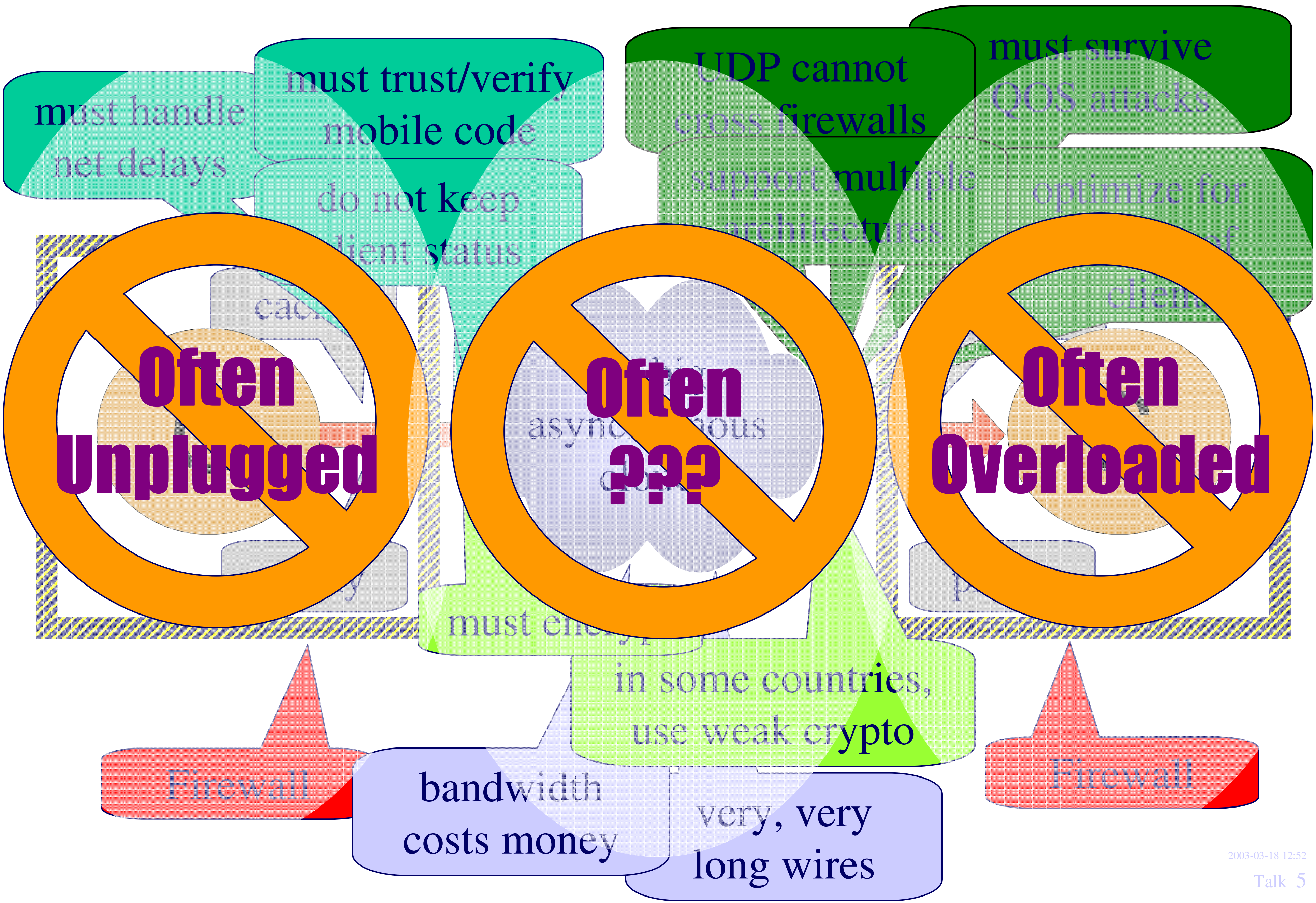
In-Memory Wires



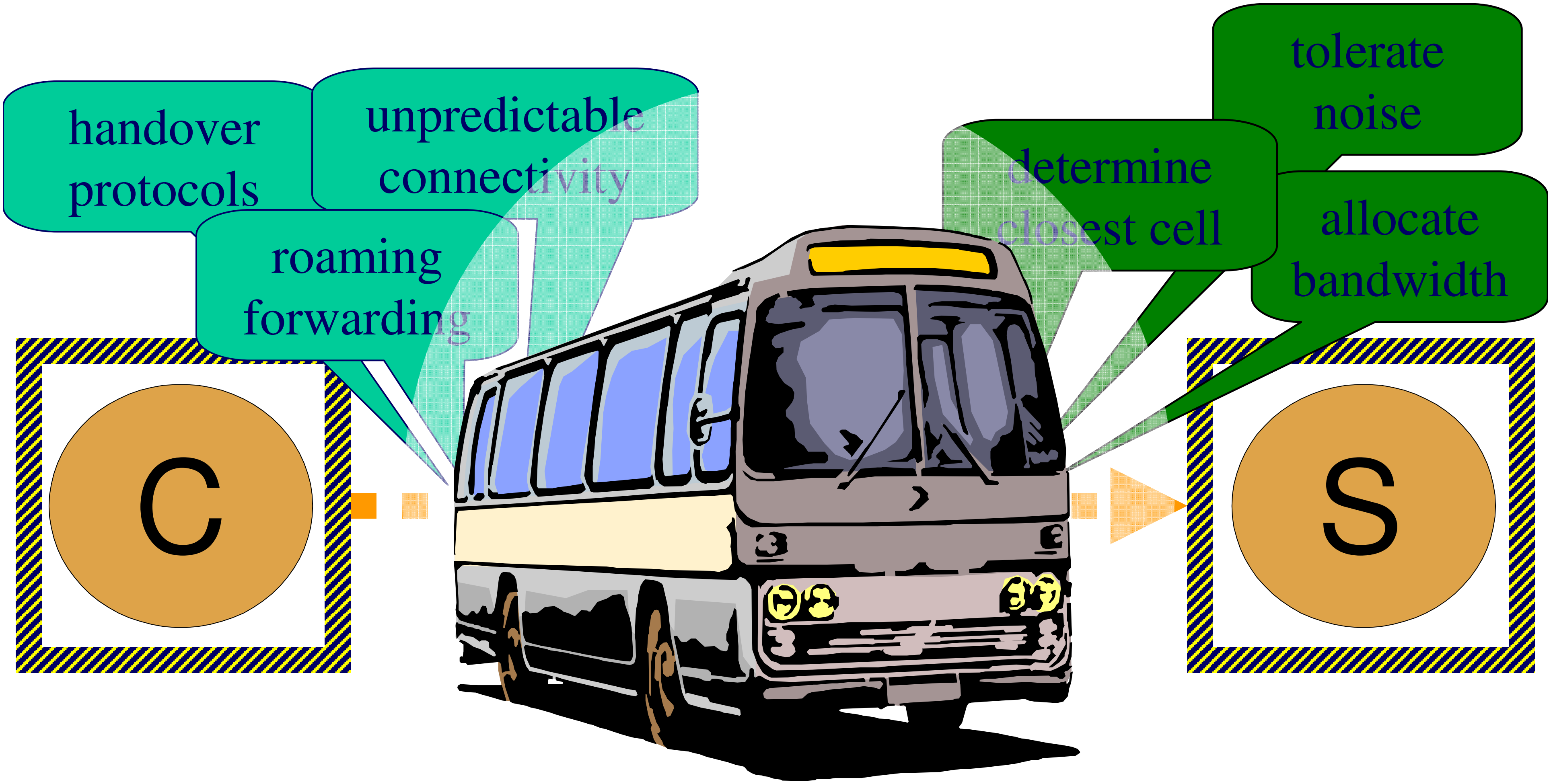
LAN Wires


must handle partial failures
audit all actions
must apply access contr
must be authenticated
keep client resource use to minimum
linearize data
relocate objects for load balancing
optimize for 1-shot access
optimize fo
proxy
proxy

WAN Wires


must handle net delays
must trust/verify mobile code
do not keep client status
UDP cannot cross firewalls
must survive QOS attacks
support multiple architectures
optimize for
Often Unplugged
Often ???
Often Overloaded
in some countries, use weak crypto
Firewall
bandwidth costs money
very, very long wires
Firewall

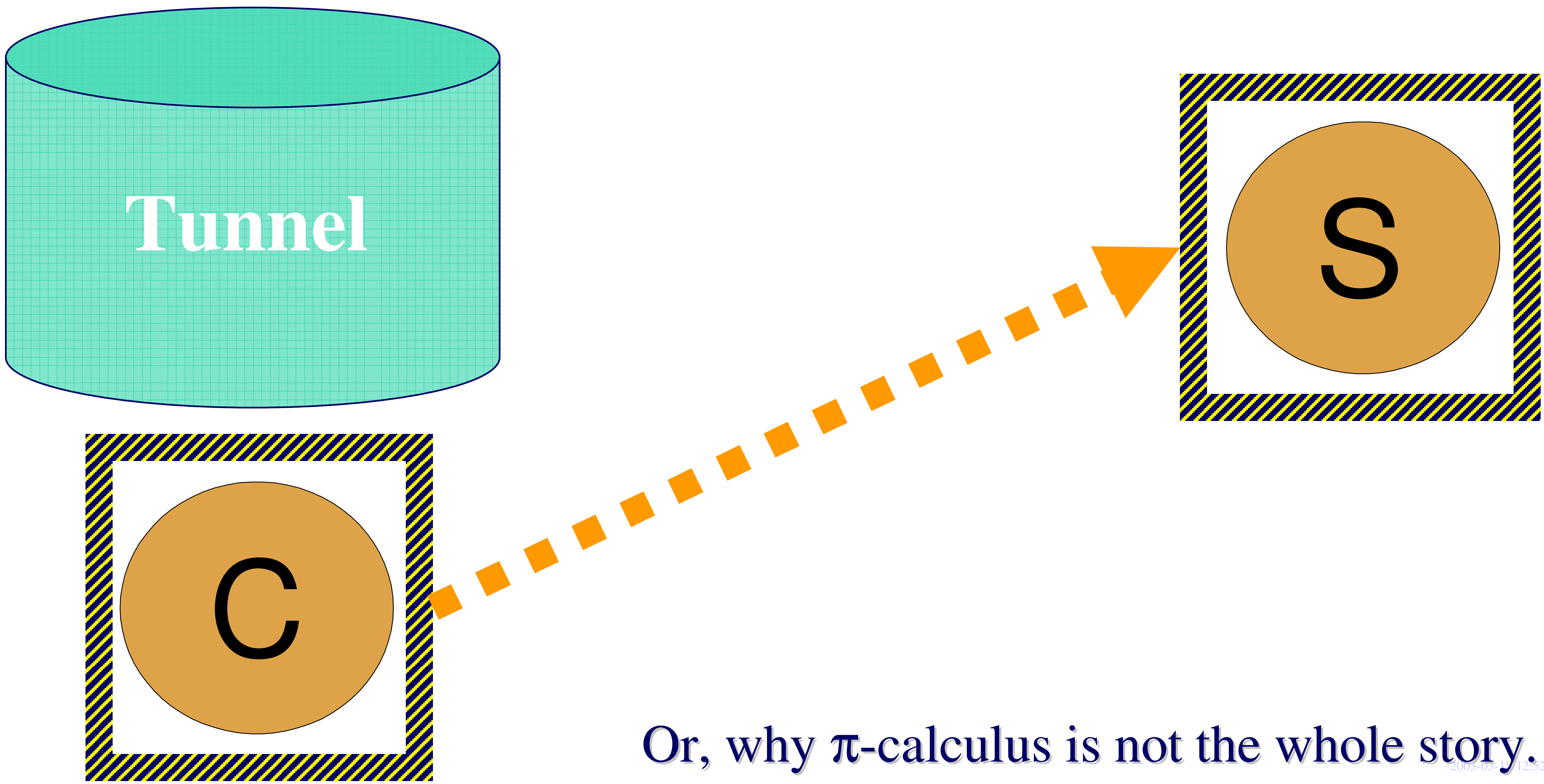
Mobile ("Wireless") Wires



Mobile obstacles

Tunnel Effect

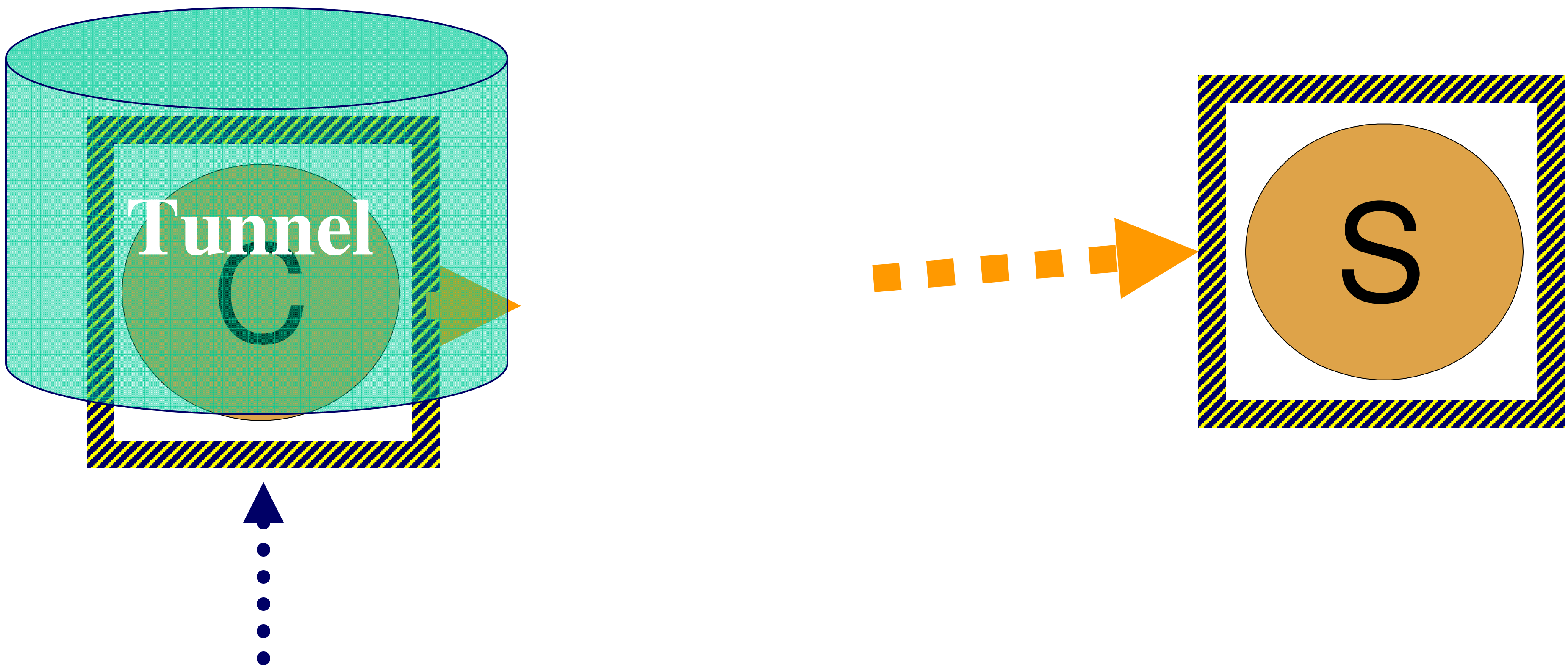
Mobile devices
going around obstacles



Or, why π-calculus is not the whole story.

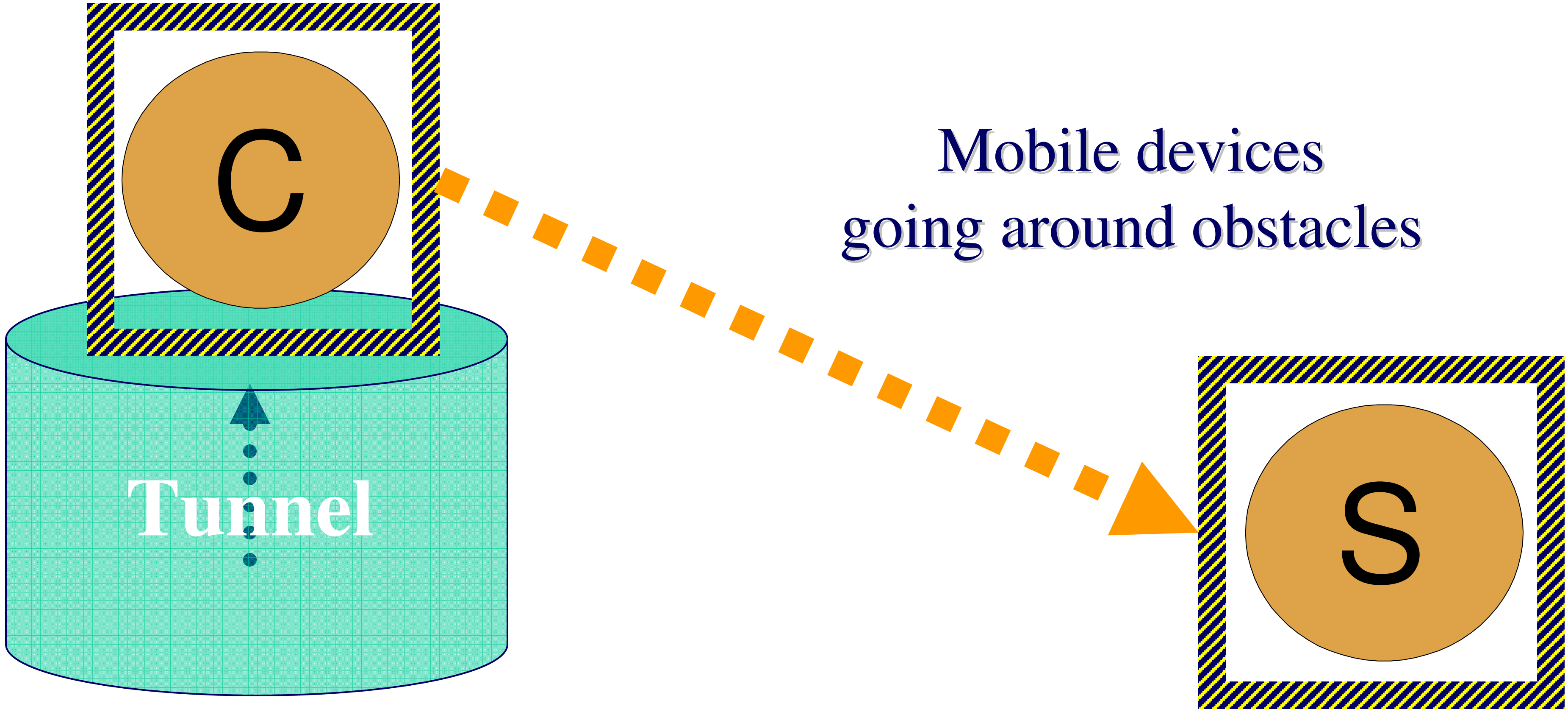
Tunnel Effect

Mobile devices
going around obstacles



Or, why π-calculus is not the whole story.

Tunnel Effect

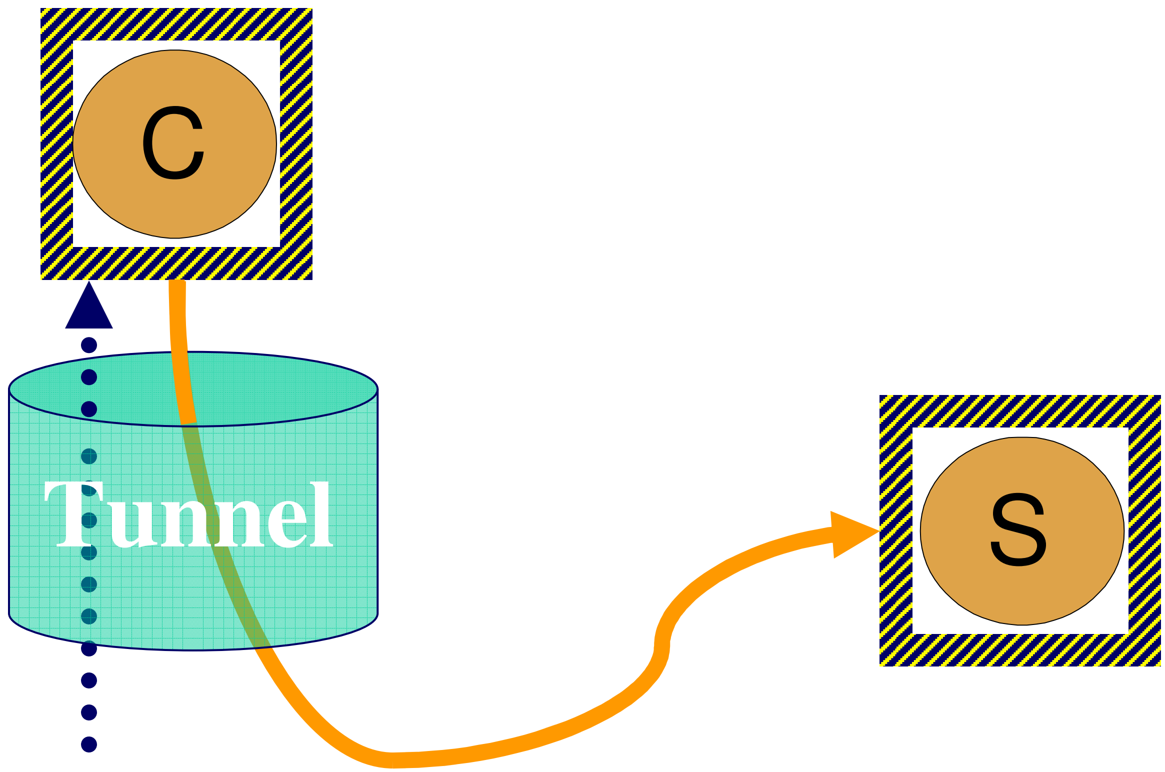


Mobile devices
going around obstacles

Or, why π-calculus is not the whole story.

Tunnels vs Reliable Communication

Reliable communication = continuous unbreakable wires



Reliable communication + Tunnels
= wires get tangled (and untangling them is hard)
= eventually one can no longer move (or the wire breaks).

About the Tunnel Effect

In hardwired communication:

- *Capable* = *Able*.
- Unless, of course, something is broken.

In the tunnel effect:

- *Capable* but *unable* to communicate.
- Moreover, nothing is broken:
  - The client is working.
  - The server is working.
  - The tunnel tunnels.
  - The ether ethers.
  - All goes back to normal without need to *fix* anything.

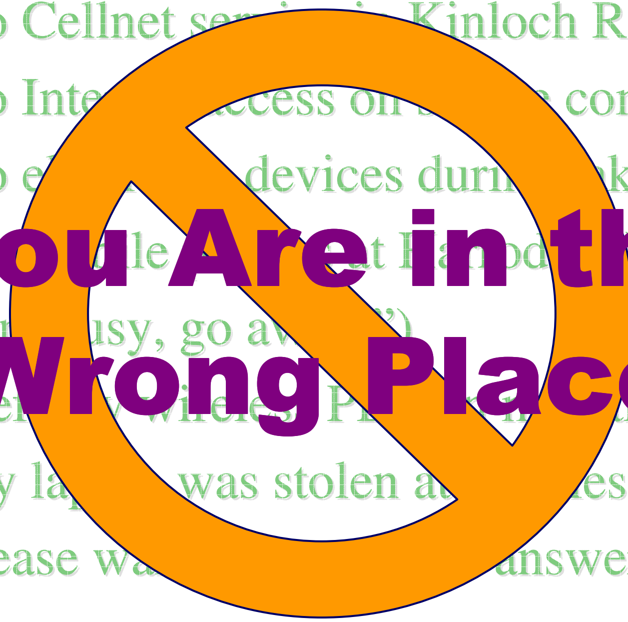
Just one of a variety of phenomena where…

Sudden Inability to Communicate

No longer to be regarded as a failure

It is a state of affairs, due to many causes:

- Congestion (“The server could not be reached.”)
- Obstructions (“Infrared device out of sight.”)
- Geography (“No Cellnet se[illegible] Kinloch Rannoch.”)
- Security (“No Inte[illegible]ccess on [illegible] computer.”)
- Safety (“No e[illegible] devices duri[illegible]keoff and landing.”)
- Policy ([illegible])
- Privacy (“I a[illegible]usy, go a[illegible]”)
- Psyche (“I fe[illegible] wireles[illegible] P[illegible] pants.”)
- Crime (“My la[illegible] was stolen a[illegible]es De Gaulle’s.”)
- Physics (“Please wa[illegible] answer from Mars.”)

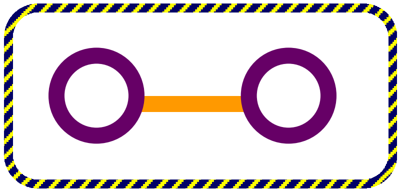


Nothing is broken

- “broken” ≜ “somebody can be found to fix the problem”.
- In the cases above, nothing is “broken”. Yet, things don’t work.
- The failure model is not “it crashed” but…

Connectivity Depends on Location

1) Proximity:



Ok. Fast (bounded delay), reliable, secure.

2) Physical distance: (possibly with virtual distance = 0)

No such thing as remote real-time control. No unbreakable links.
Observationally different from (1).

3) Virtual distance: (possibly with physical distance = 0)

No such thing as implicitly secure remote links.
Observationally different from (1).

Global Computation

How do we embed the features and restrictions of global communication in a computational model?

Must abandon function call/handshake.

- We cannot afford to have every function call over the network to block waiting for an answer. (π vs. async-π.)

Must abandon symmetric multi-party (even async) communication.

- We cannot afford to solve consensus problems all the time. (async-π vs. join.)

Must abandon pointers/references.

- We cannot afford references of any kind that are *always* connected to their target, and we must be able to reconnect them. (π vs. ambients.)

Must abandon familiar failure models.

- We cannot assume that every failure leads to an exception.
- We cannot assume we are even allowed to know that a failure ever happened.

Ambients Approach

We want to capture in an abstract way, notions of locality, of mobility, and of ability to cross barriers.

An *ambient* is a place, delimited by a boundary, where computation happens.

Ambients have a name, a collection of local processes, and a collection of subambients.

Ambients can move in an out of other ambients, subject to capabilities that are associated with ambient names.

Ambient names are unforgeable (as in π and spi).

The Ambient Calculus

The *Ambient Calculus*: a computational model for:

- Behaviors that are *capable* but sometimes *unable* to communicate.
- Communication that is neither *broken* nor *not broken*.

Spatial structures (agents, networks, etc.) are represented by nested locations:

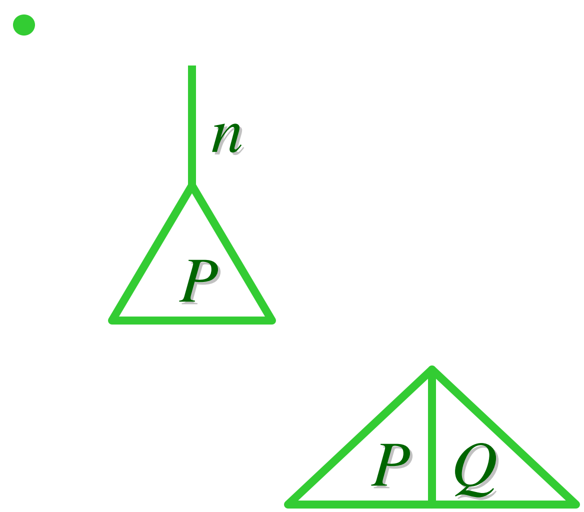
Processes

$\mathbf{0}$ (void)

$n[P]$ (location)

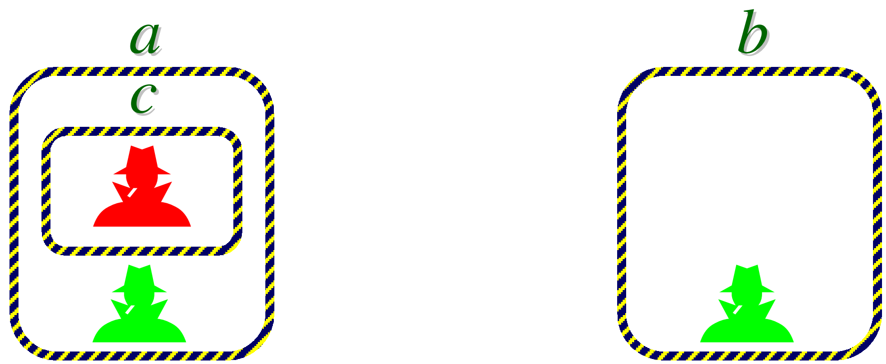
$P \mid Q$ (composition)

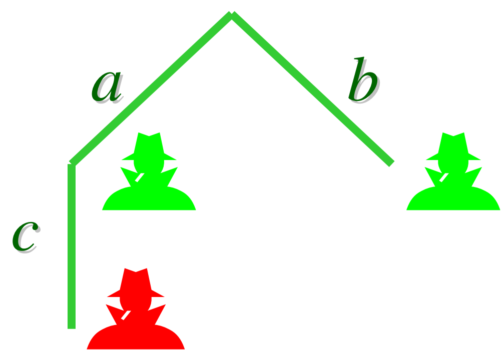
Tree Representation



Mobility

Mobility is change of spatial structures over time.

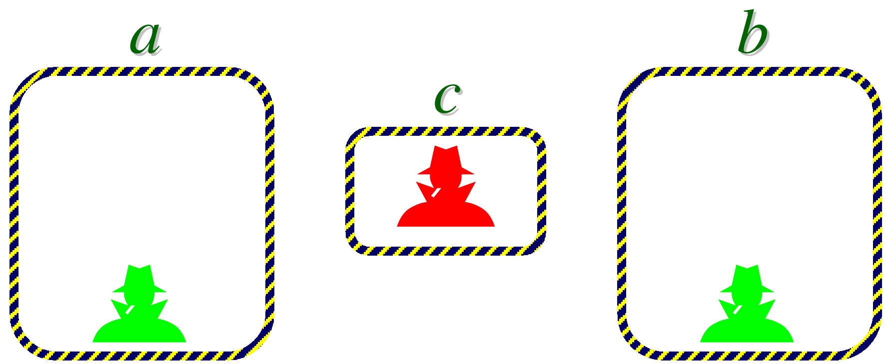


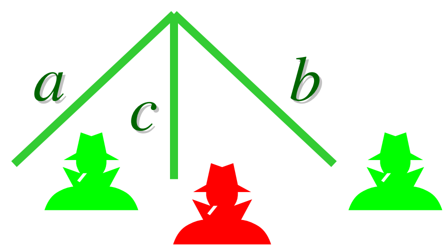


$$a[Q \mid c[out\ a.\ in\ b.\ P]] \qquad \mid b[R]$$

Mobility

Mobility is change of spatial structures over time.

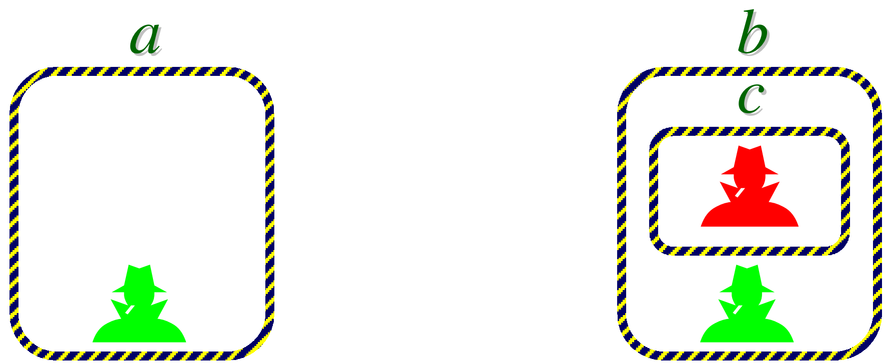


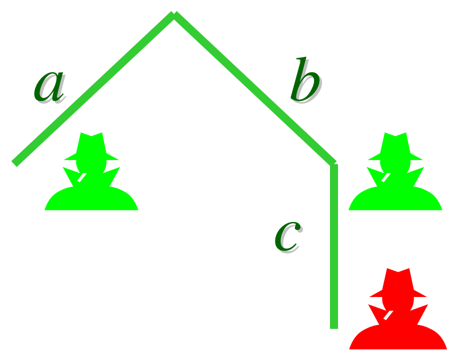


$a[Q] \quad | \; c[in\ b.\ P] \; | \; b[R]$

Mobility

Mobility is change of spatial structures over time.



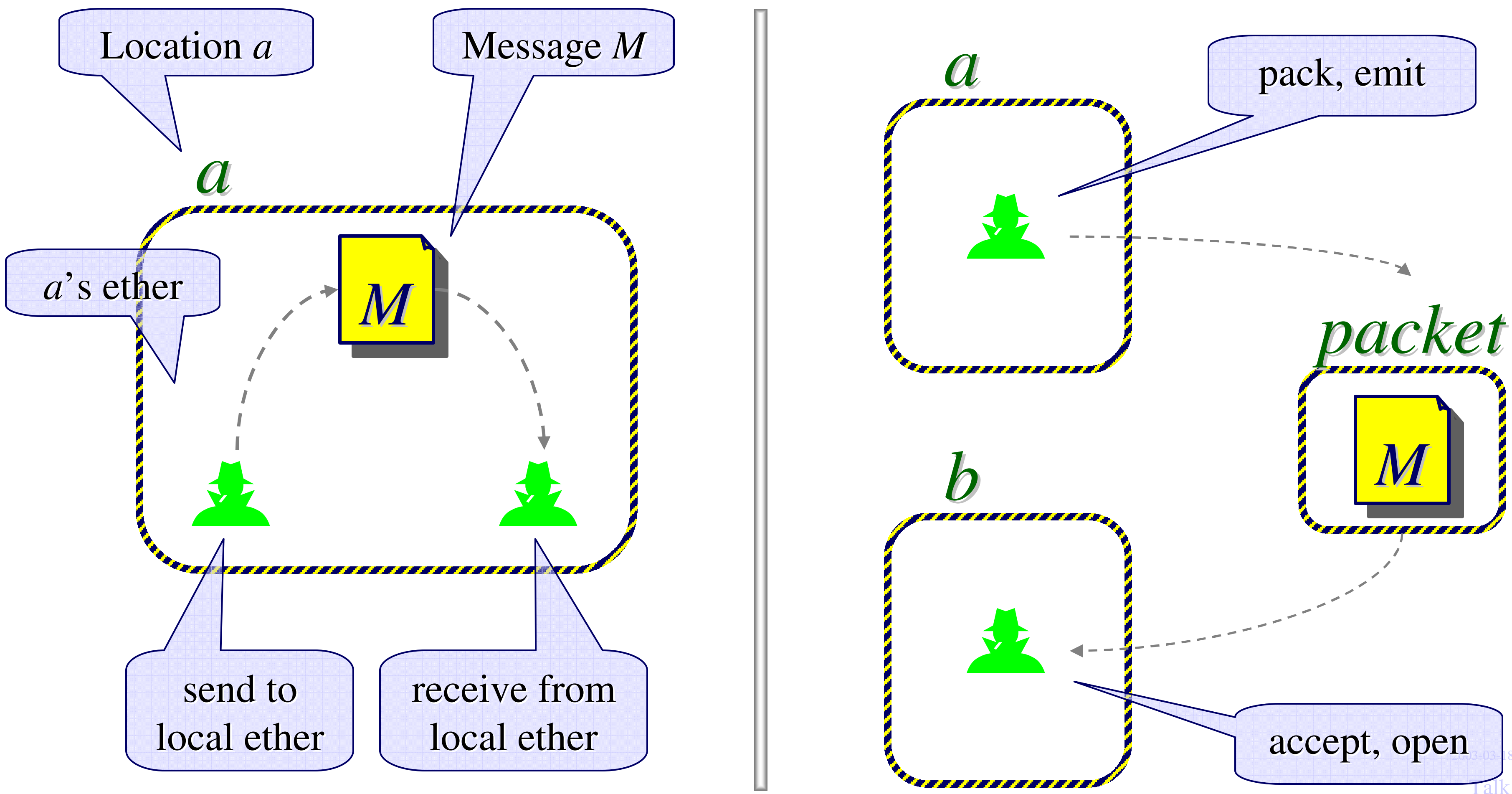


$a[Q] \mid b[R \mid c[P]]$

Communication

Communication is strictly local, within a given location.

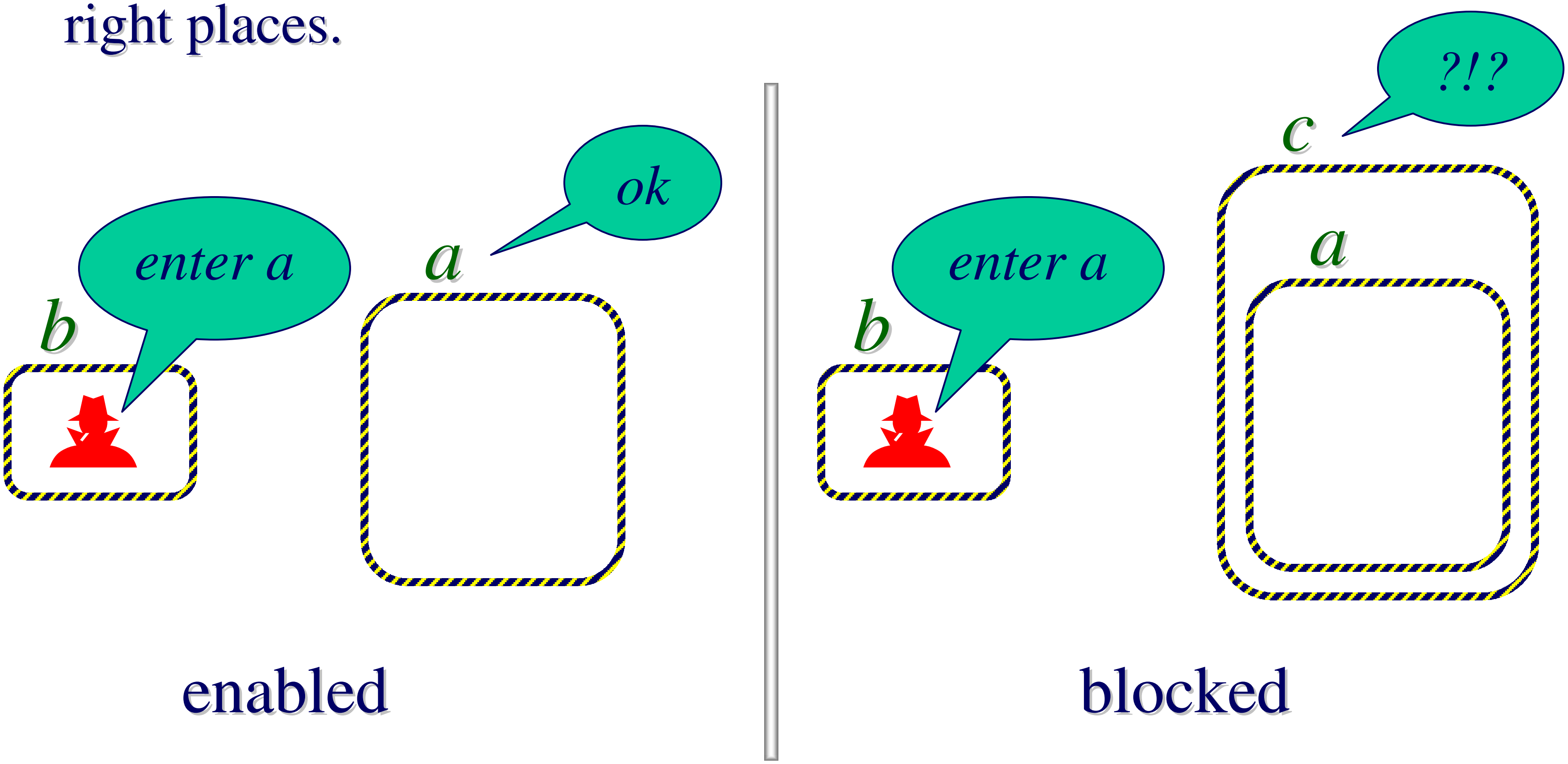
Remote communication must be simulated by sending around mobile packets (which may get lost).



Security

Security issues are reduced to the ability to create, destroy, enter and exit locations.

- π-calculus restriction accounts for private capabilities.
- As for communication, capabilities can be exercised only in the right places.



The Ambient Calculus

$P \in \Pi ::=$	Processes		$M ::=$	Messages
$(\nu n)P$	restriction	Location Trees *Spatial*	n	name
$\mathbf{0}$	inactivity		*in* M	entry capability
$P \mid P'$	parallel		*out* M	exit capability
$M[P]$	ambient		*open* M	open capability
$!P$	replication		ε	empty path
$M.P$	exercise a capability	Actions *Temporal*	$M.M'$	composite path
$(n).P$	input locally, bind to n			
$\langle M \rangle$	output locally (async)			

$n[] \triangleq n[\mathbf{0}]$

$M \triangleq M.\mathbf{0}$ (where appropriate)

Reduction Semantics

A structural congruence relation $P \equiv Q$:

- On spatial expressions, $P \equiv Q$ iff P and Q denote the same tree. So, the syntax modulo $\equiv$ is a notation for spatial trees.
- On full ambient expressions, $P \equiv Q$ if in addition the respective threads are "trivially equivalent".
- Prominent in the definition of the logic.

A reduction relation $P \longrightarrow^* Q$:

- Defining the meaning of mobility and communication actions.
- Closed up to structural congruence:

$$P \equiv P', \; P' \longrightarrow^* Q', \; Q' \equiv Q \quad \Rightarrow \quad P \longrightarrow^* Q$$

Reduction

$n[\mathit{in}\ m.\ P \mid Q] \mid m[R]$	$\longrightarrow m[n[P \mid Q] \mid R]$	(Red In)
$m[n[\mathit{out}\ m.\ P \mid Q] \mid R]$	$\longrightarrow n[P \mid Q] \mid m[R]$	(Red Out)
$\mathit{open}\ m.\ P \mid m[Q]$	$\longrightarrow P \mid Q$	(Red Open)
$(n).P \mid \langle M \rangle$	$\longrightarrow P\{n \leftarrow M\}$	(Red Comm)
$P \longrightarrow Q \;\Rightarrow\; (\nu n)P \longrightarrow (\nu n)Q$		(Red Res)
$P \longrightarrow Q \;\Rightarrow\; n[P] \longrightarrow n[Q]$		(Red Amb)
$P \longrightarrow Q \;\Rightarrow\; P \mid R \longrightarrow Q \mid R$		(Red Par)
$P' \equiv P,\ P \longrightarrow Q,\ Q \equiv Q' \;\Rightarrow\; P' \longrightarrow Q'$		(Red $\equiv$)

$\longrightarrow^*$ is the reflexive-transitive closure of $\longrightarrow$

Structural Congruence

$P \equiv P$	(Struct Refl)
$P \equiv Q \;\Rightarrow\; Q \equiv P$	(Struct Symm)
$P \equiv Q, Q \equiv R \;\Rightarrow\; P \equiv R$	(Struct Trans)
$P \equiv Q \;\Rightarrow\; (\nu n)P \equiv (\nu n)Q$	(Struct Res)
$P \equiv Q \;\Rightarrow\; P \mid R \equiv Q \mid R$	(Struct Par)
$P \equiv Q \;\Rightarrow\; !P \equiv !Q$	(Struct Repl)
$P \equiv Q \;\Rightarrow\; M[P] \equiv M[Q]$	(Struct Amb)
$P \equiv Q \;\Rightarrow\; M.P \equiv M.Q$	(Struct Action)
$P \equiv Q \;\Rightarrow\; (n).P \equiv (n).Q$	(Struct Input)
$\varepsilon.P \equiv P$	(Struct ε)
$(M.M').P \equiv M.M'.P$	(Struct .)

$(\nu n)\mathbf{0} \equiv \mathbf{0}$	(Struct Res Zero)
$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$	(Struct Res Res)
$(\nu n)(P \mid Q) \equiv P \mid (\nu n)Q \quad \text{if } n \notin fn(P)$	(Struct Res Par)
$(\nu n)(m[P]) \equiv m[(\nu n)P] \quad \text{if } n \neq m$	(Struct Res Amb)
$P \mid Q \equiv Q \mid P$	(Struct Par Comm)
$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$	(Struct Par Assoc)
$P \mid \mathbf{0} \equiv P$	(Struct Par Zero)
$\mathbf{!}(P \mid Q) \equiv \mathbf{!}P \mid \mathbf{!}Q$	(Struct Repl Par)
$\mathbf{!0} \equiv \mathbf{0}$	(Struct Repl Zero)
$\mathbf{!}P \equiv P \mid \mathbf{!}P$	(Struct Repl Copy)
$\mathbf{!}P \equiv \mathbf{!!}P$	(Struct Repl Repl)

These axioms (particularly the ones for **!**) are sound and complete with respect to equality of spatial trees: edge-labeled finite-depth unordered trees, with infinite-branching but finitely many distinct labels under each node.

Ambient Calculus: Example

$$\overbrace{a[msg[\langle M\rangle \mid \underbrace{out\ a.\ in\ b]]}_{\textit{send M from a to b}}}^{\textit{location a}} \mid \overbrace{b[\underbrace{open\ msg.\ (n).\ P}_{\textit{receive n; do P}}]}^{\textit{location b}}$$

The packet *msg* moves from *a* to *b*, mediated by the capabilities *out a* (to exit *a*), *in b* (to enter *b*), and *open msg* (to open the *msg* envelope).

		$a[msg[\langle M\rangle \mid out\ a.\ in\ b]]$	$\mid b[open\ msg.\ (n).\ P]$
(exit)	$\longrightarrow$	$a[] \mid msg[\langle M\rangle \mid in\ b]$	$\mid b[open\ msg.\ (n).\ P]$
(enter)	$\longrightarrow$	$a[]$	$\mid b[msg[\langle M\rangle] \mid open\ msg.\ (n).\ P]$
(open)	$\longrightarrow$	$a[]$	$\mid b[\langle M\rangle \mid (n).\ P]$
(read)	$\longrightarrow$	$a[]$	$\mid b[P\{n\leftarrow M\}]$

Noticeable Inequivalences

Replication creates new names:

$$!(\nu n)P \quad \not\equiv \quad (\nu n)!P$$

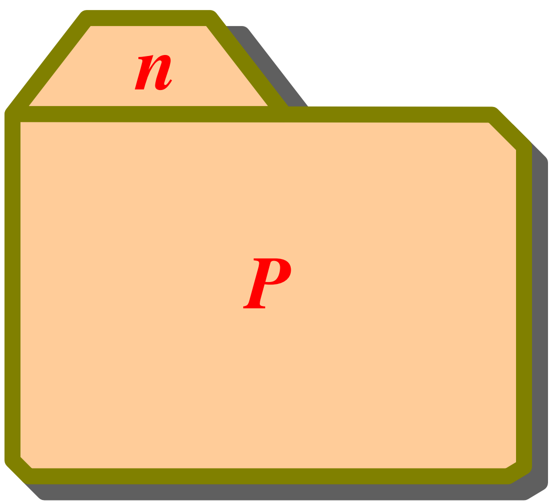
Multiple n ambients have separate identity:

$$n[P] \mid n[Q] \quad \not\equiv \quad n[P \mid Q]$$

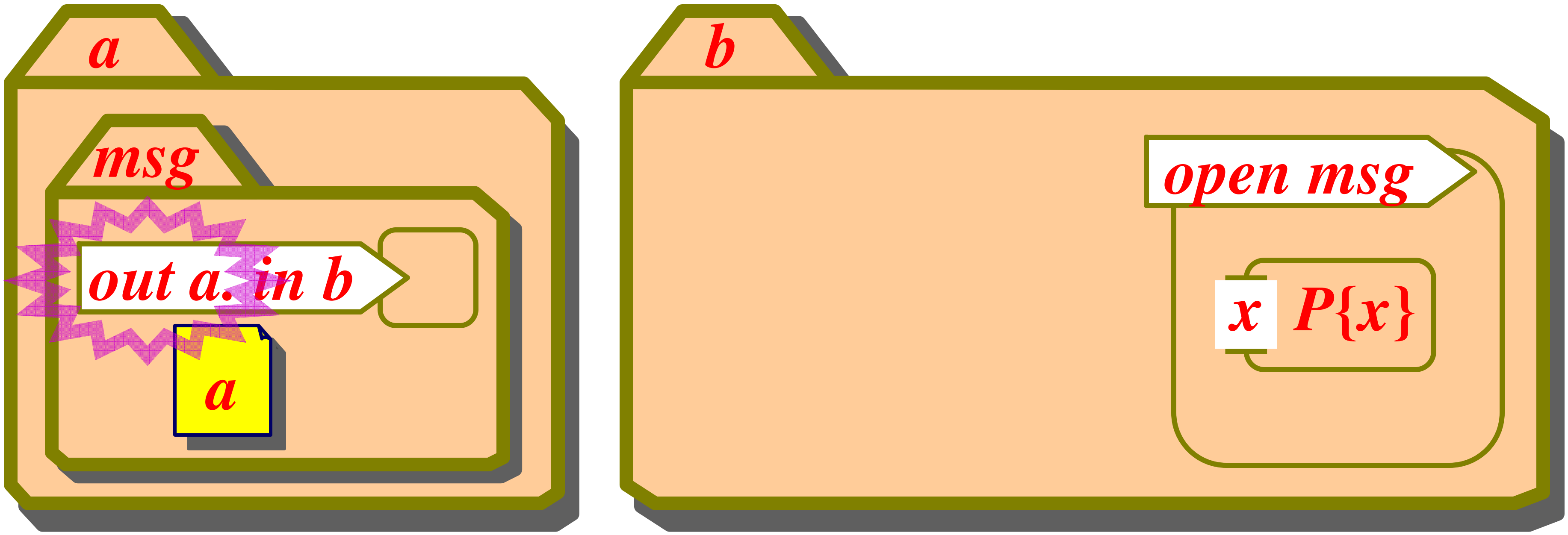
Folder Metaphor

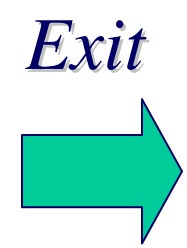
An ambient can be graphically represented as a folder:

- Consisting of a folder name *n*,
- And active contents *P*, including:
  - Hierarchical data, and computations ("gremlins").
  - Primitives for mobility and communication.

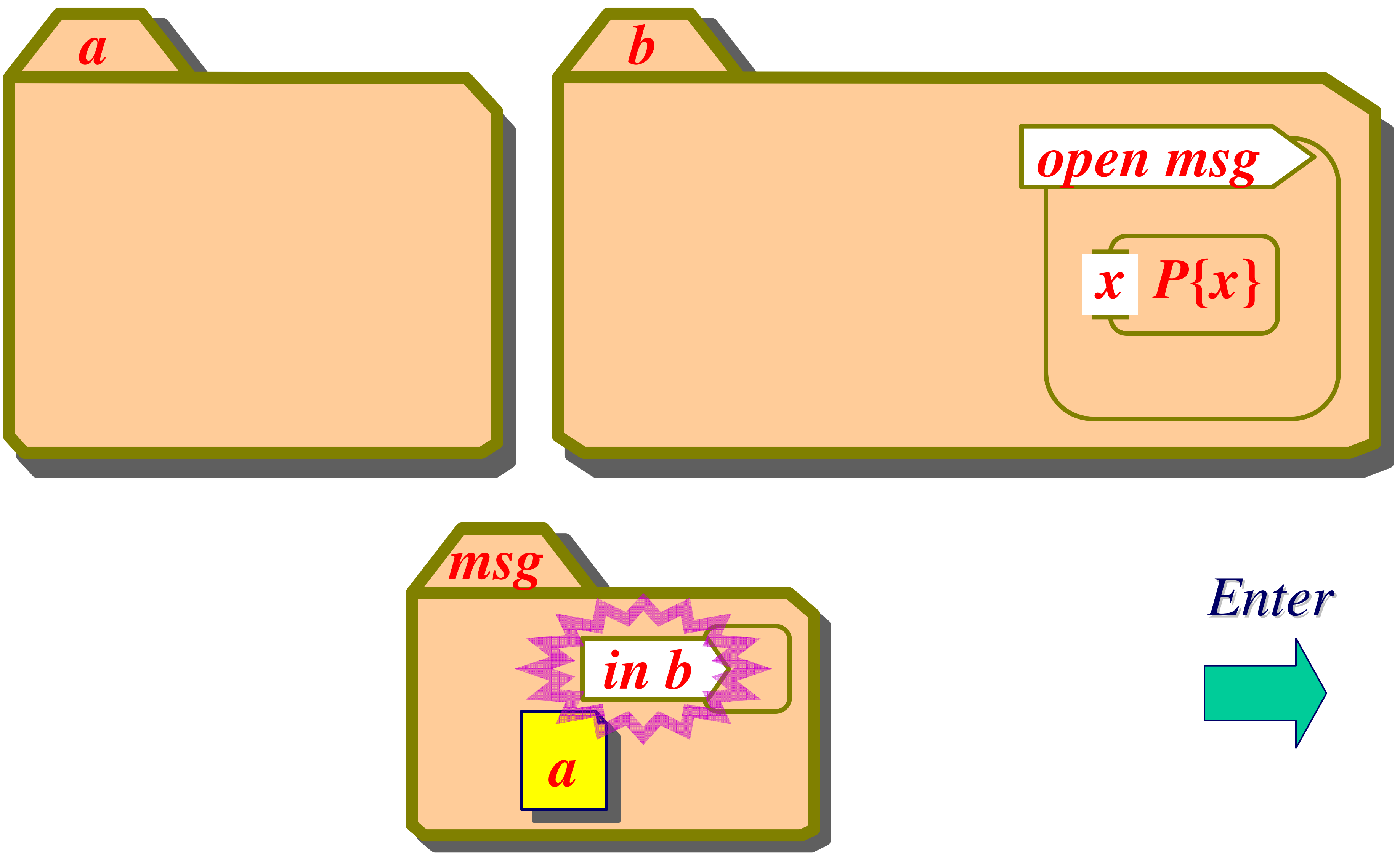


Example: Message from *a* to *b*

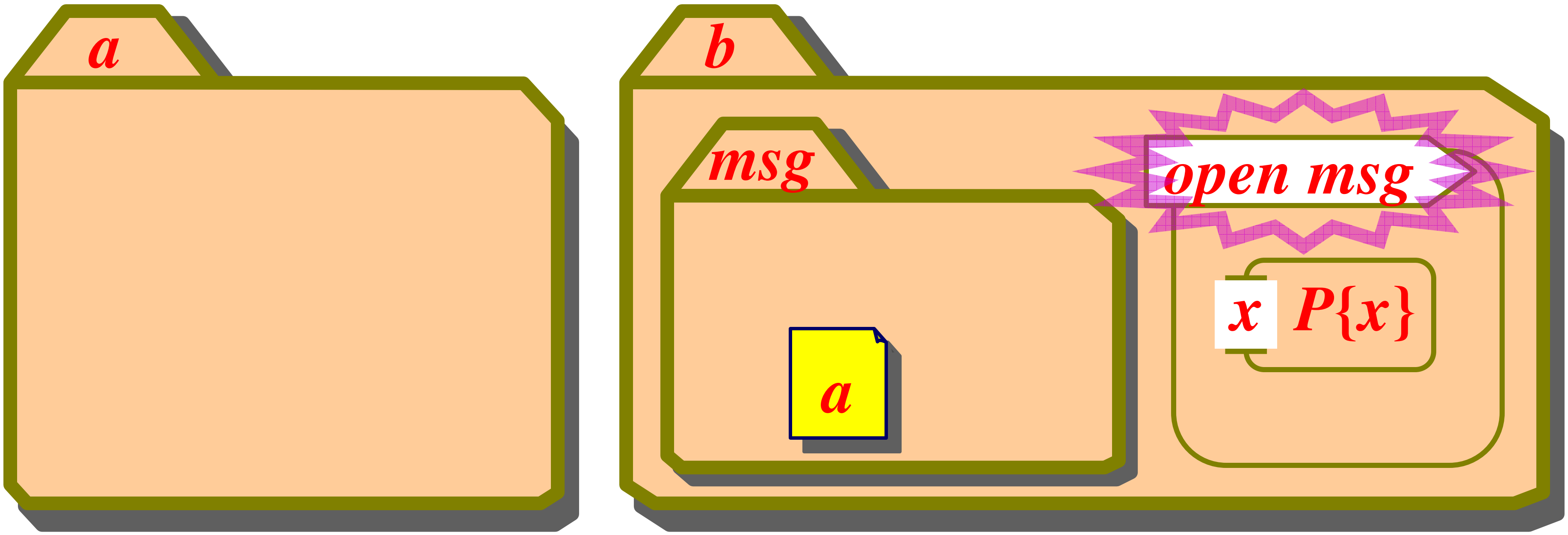




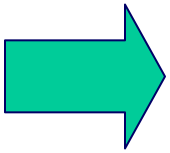
Example: Message from *a* to *b*



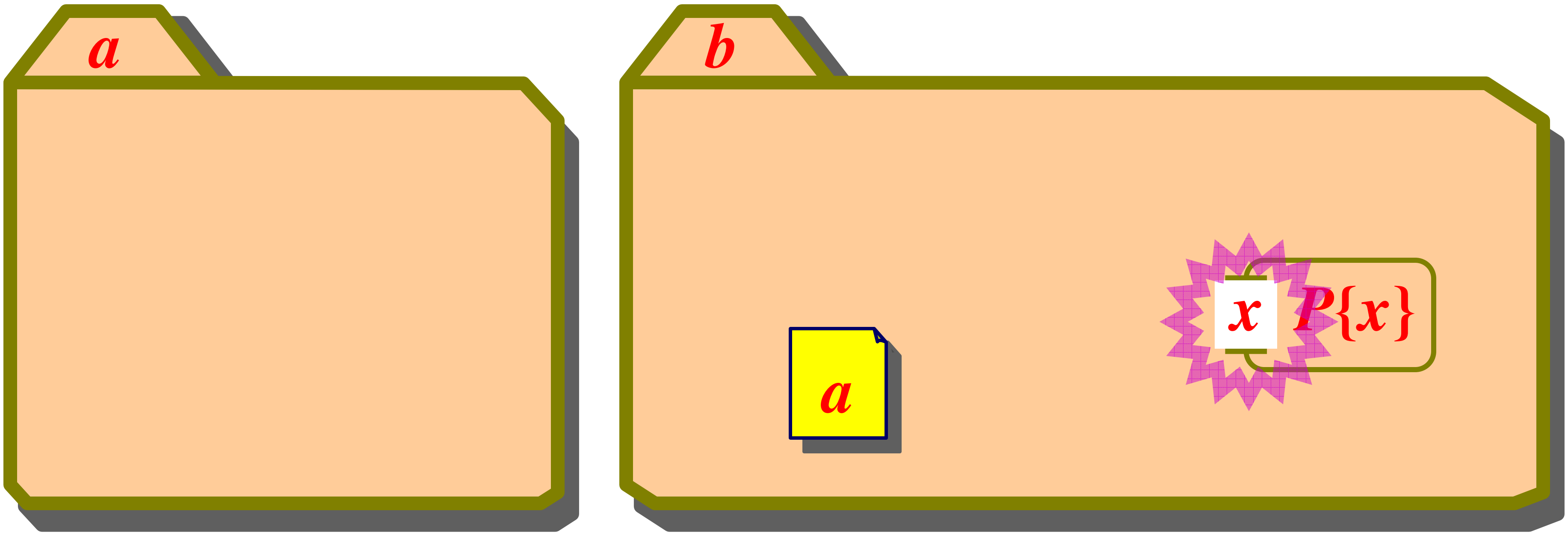
Example: Message from *a* to *b*

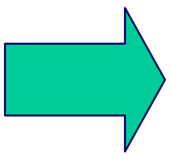


Open

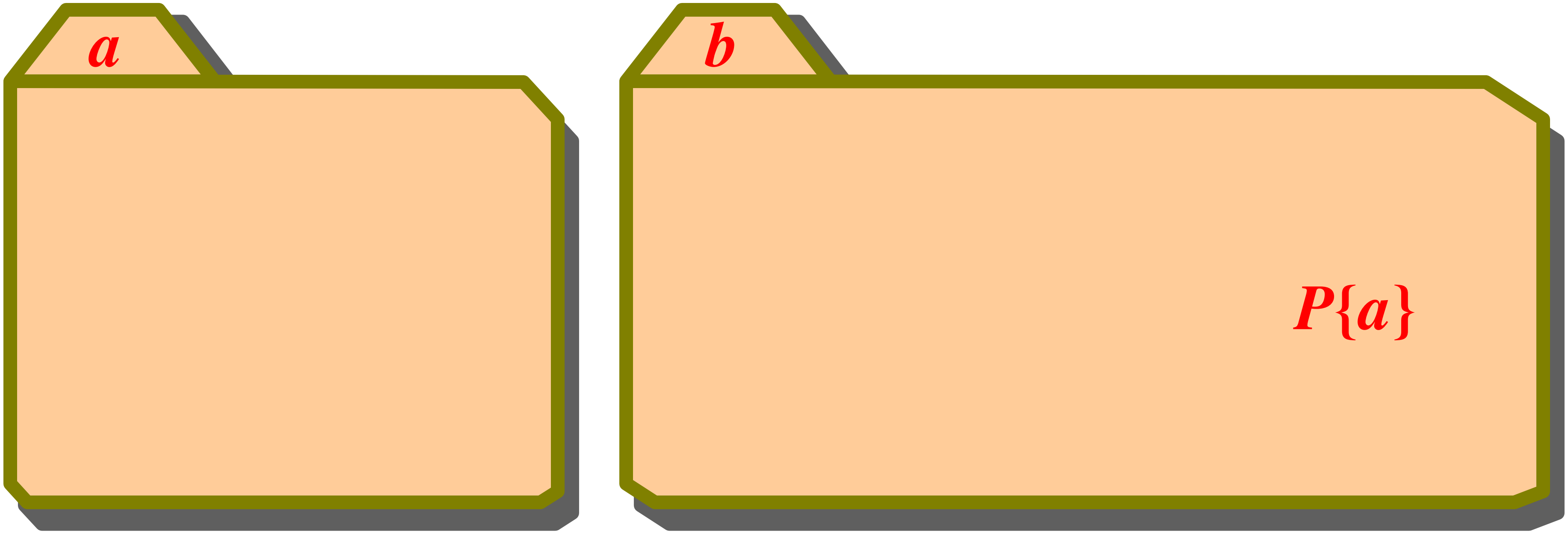


Example: Message from *a* to *b*

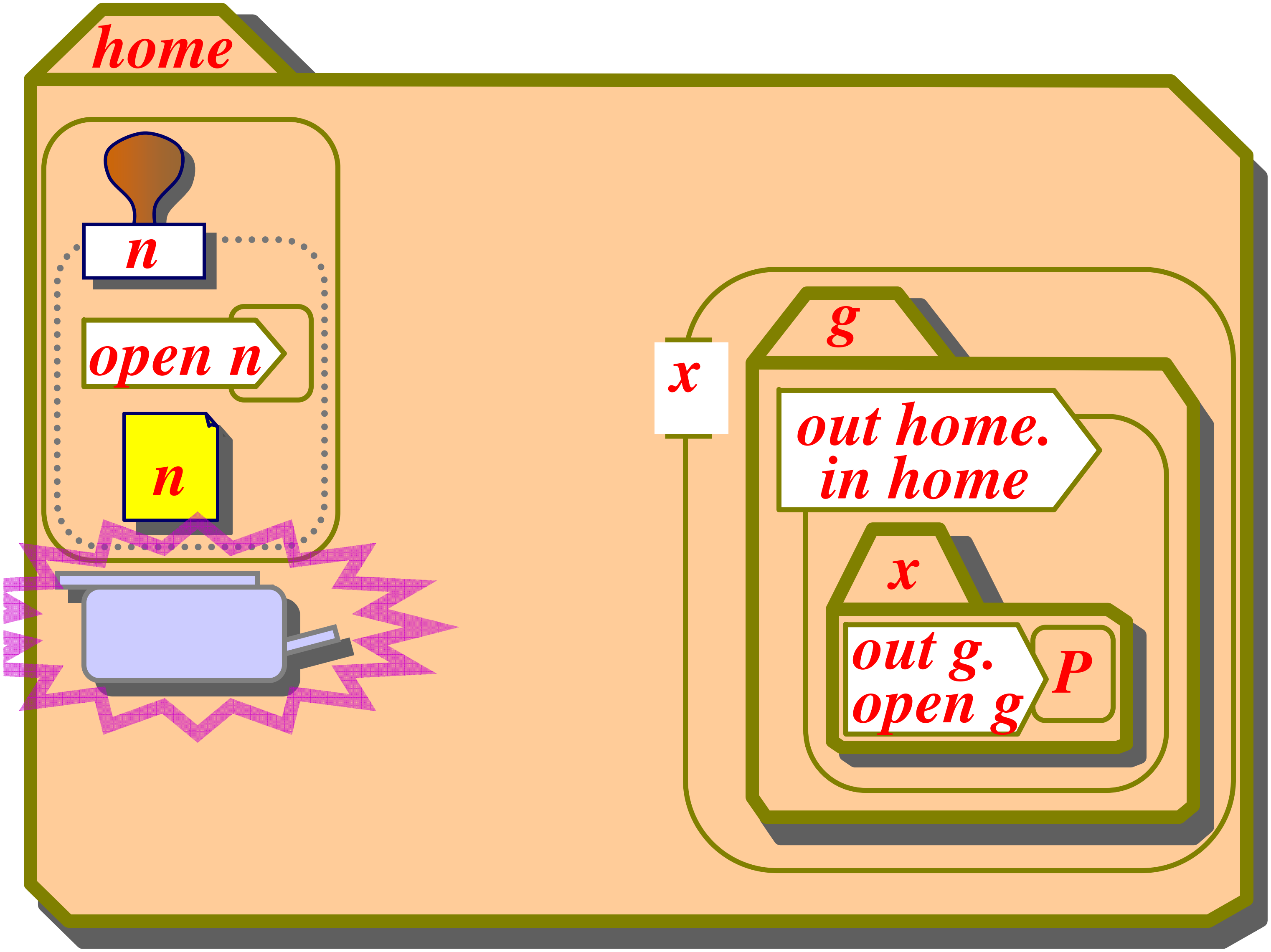




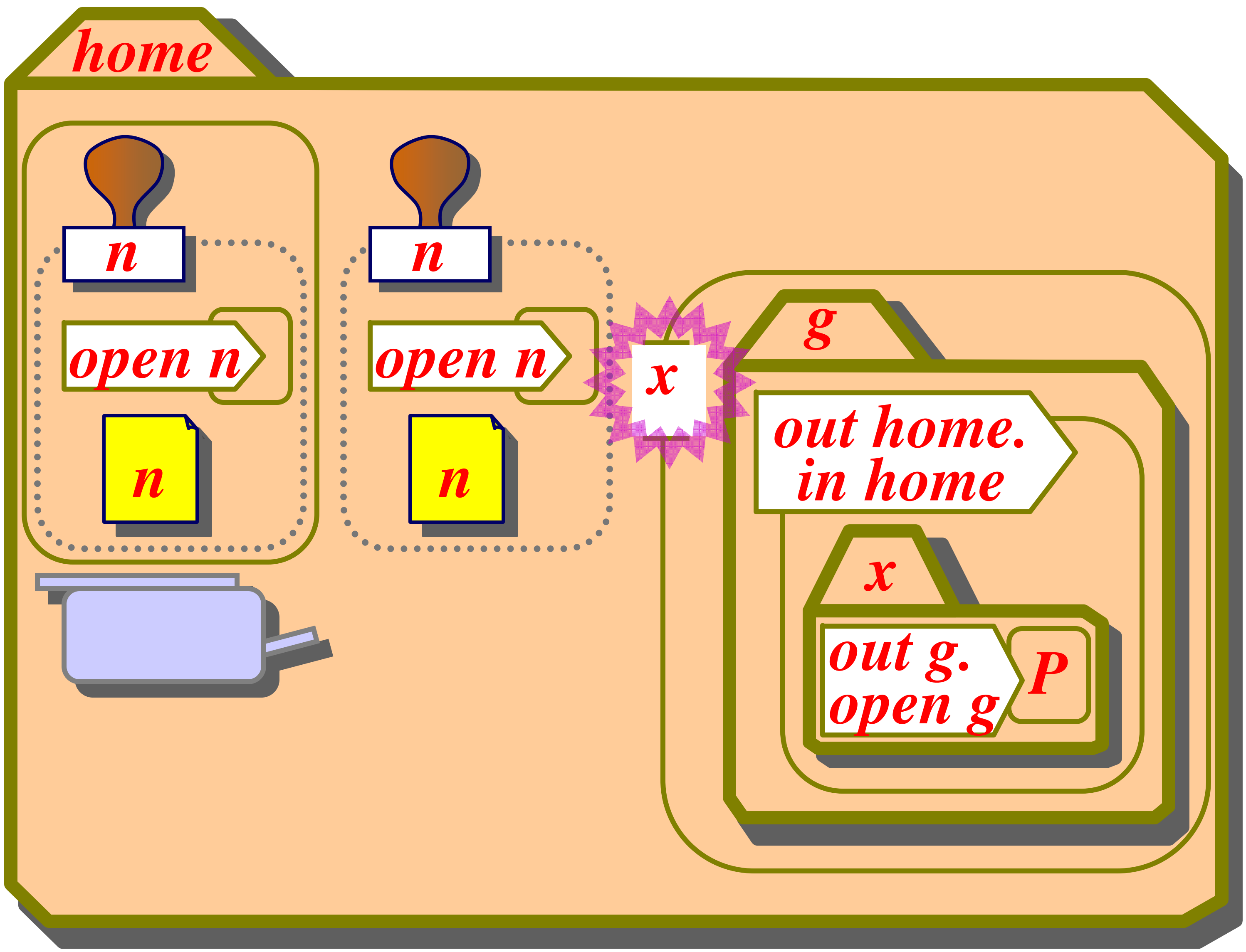
Example: Message from a to b



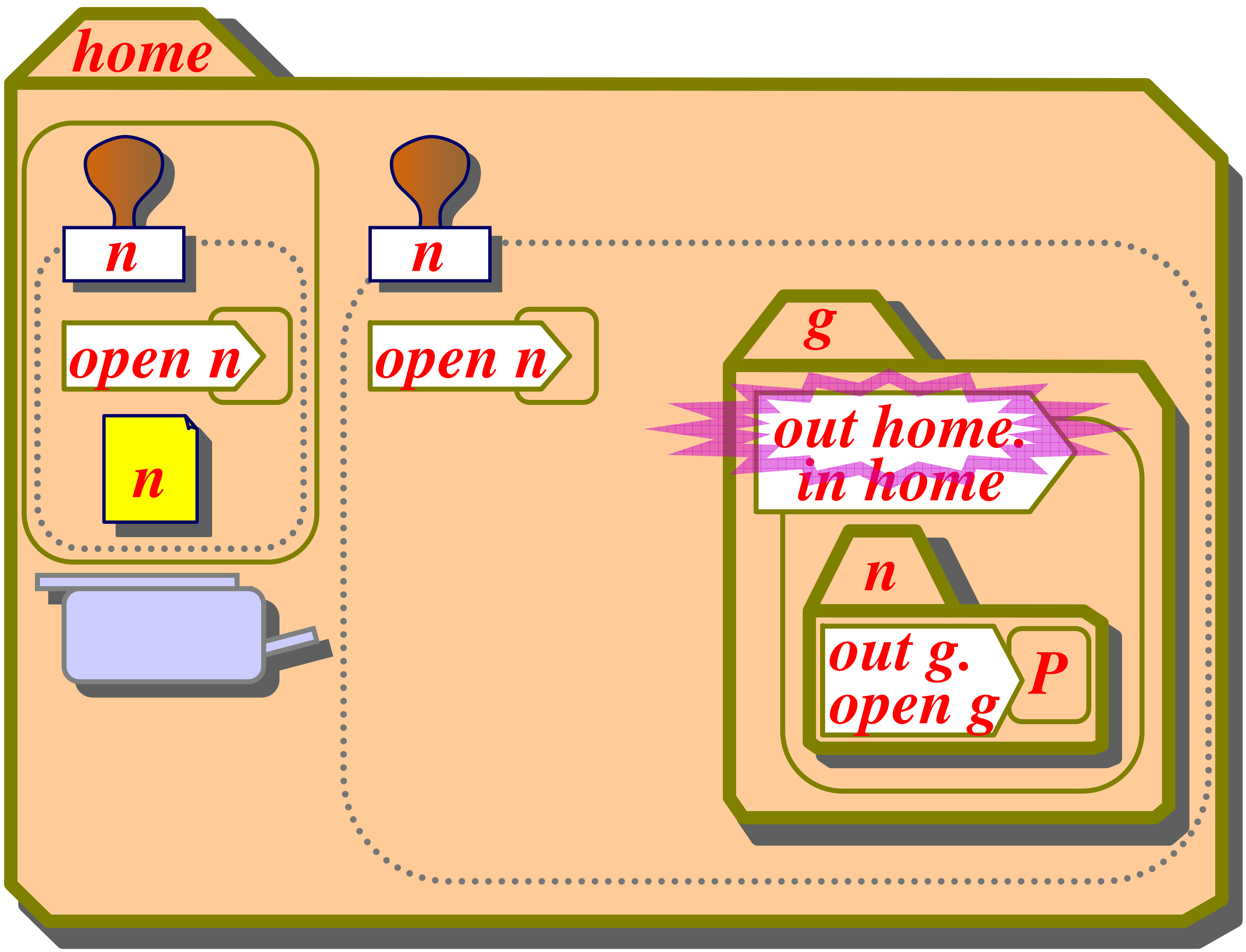
Example: Agent Authentication



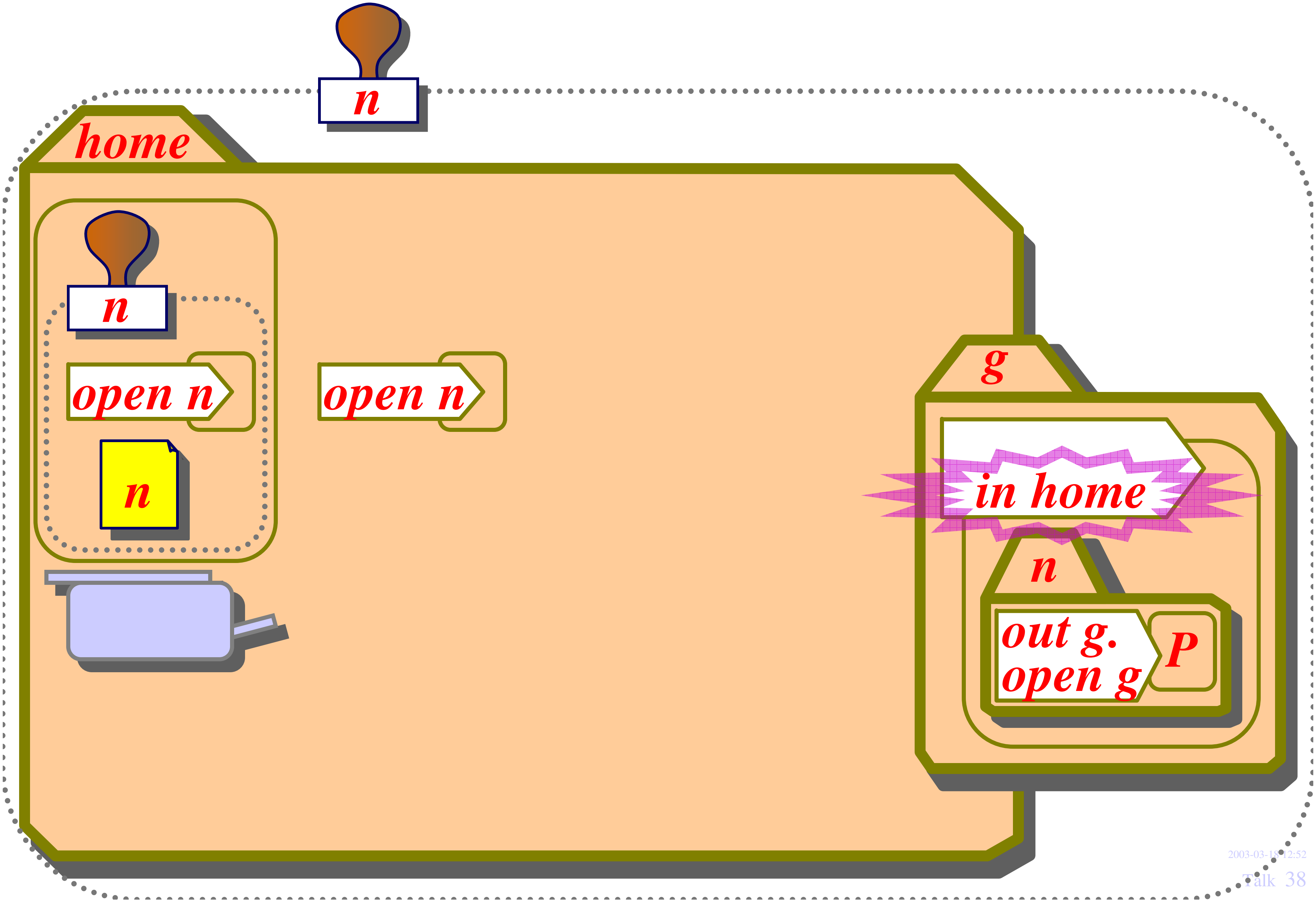
Example: Agent Authentication



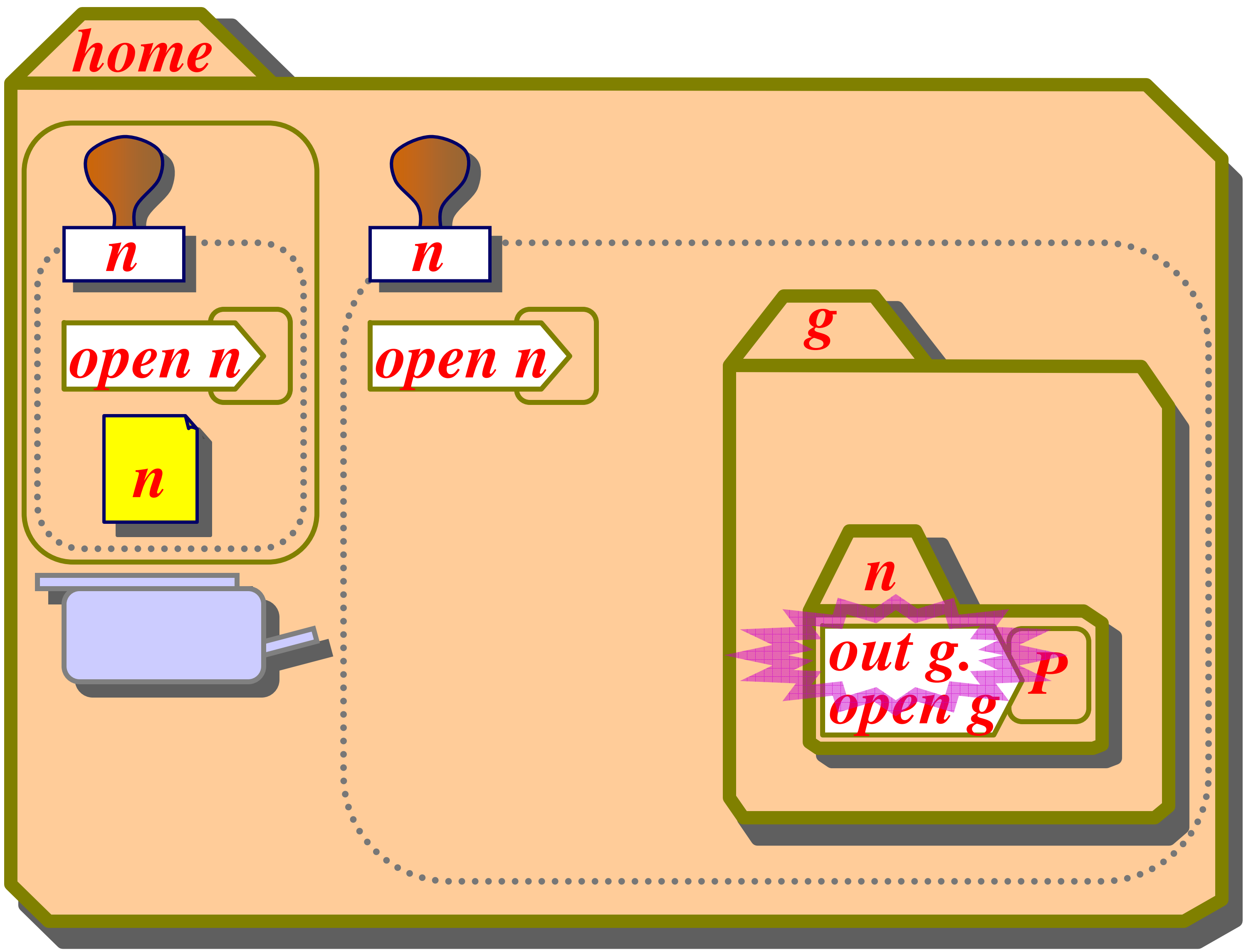
Example: Agent Authentication



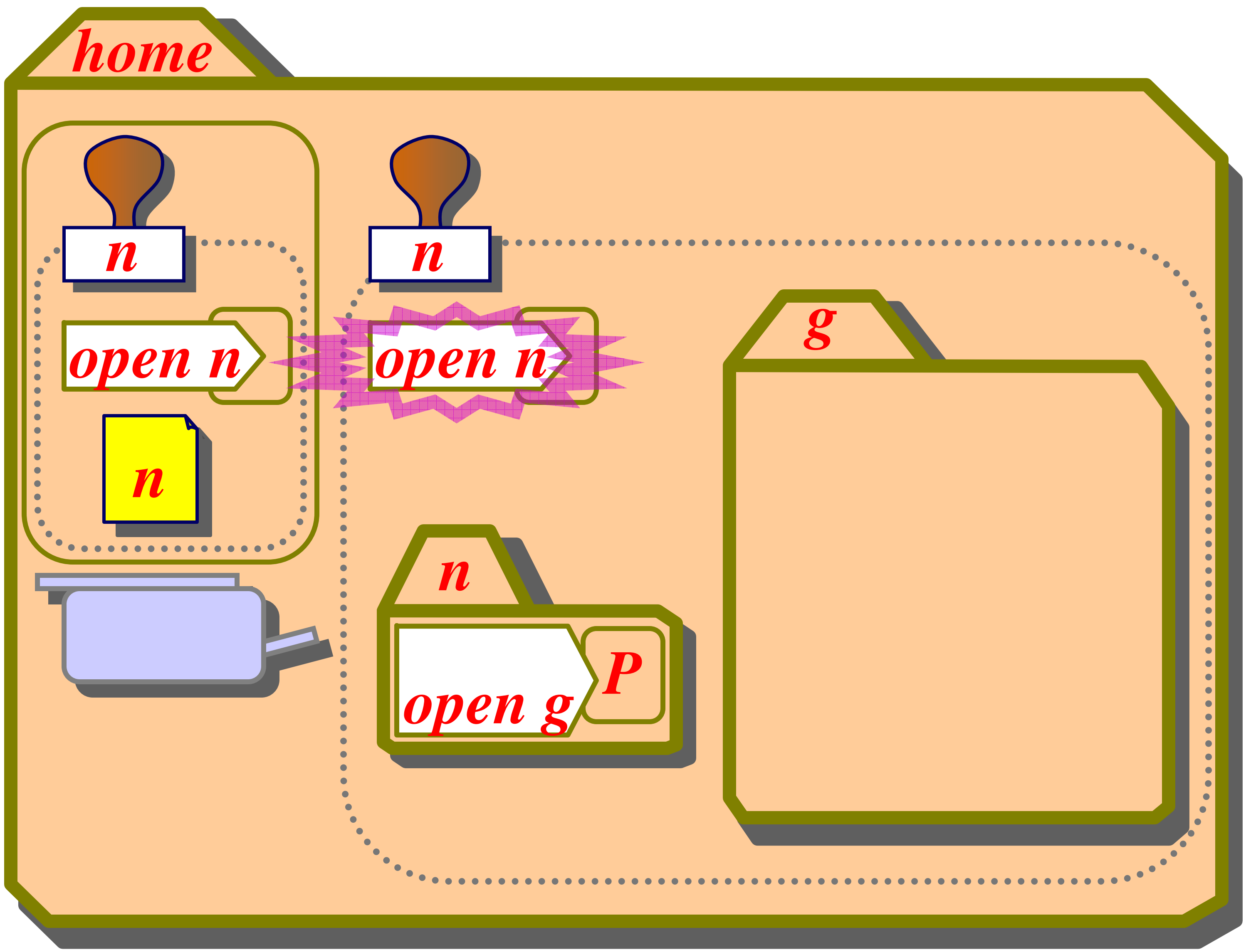
Example: Agent Authentication



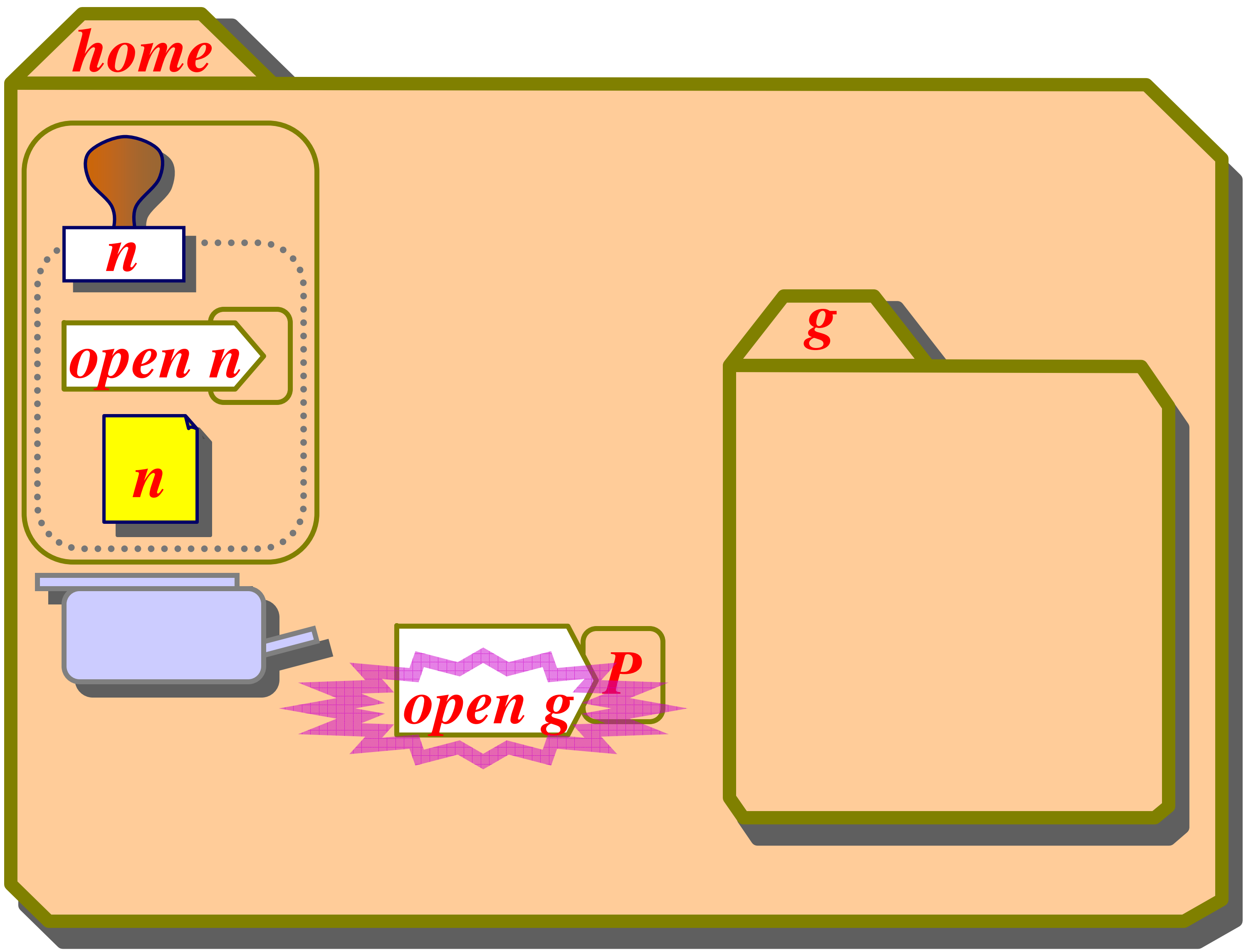
Example: Agent Authentication



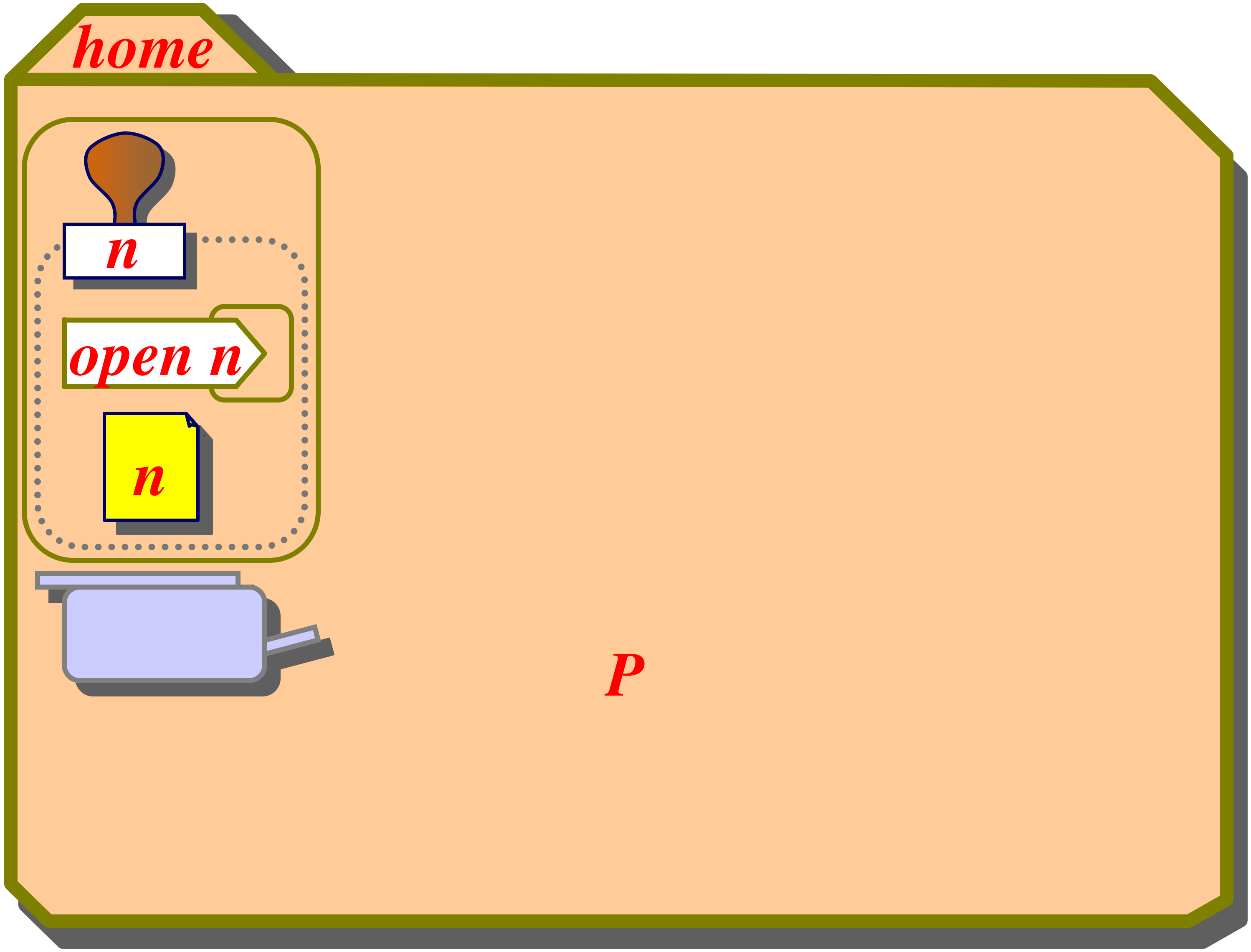
Example: Agent Authentication



Example: Agent Authentication



Example: Agent Authentication



Calculi for Communication

One basic notion

- Communication channels (a.k.a. *wires*).

One billion variations

- Value passing / name passing / process passing
- Synchronous / asynchronous / broadcast
- Internal choice / external choice / mixed choice / no choice
- Linearity / fresh output
- …

Calculi for Mobility

One basic notion

- Dynamic topology

One million variations

- Name mobility, process mobility
- Synchronous / asynchronous / datagram
- Actions / coactions / intermediaries
- Talk to local ether / talk to parent / talk to children
- …

Safe Ambients [Levi, Sangiorgi]

"*Each action has an equal and opposite coaction.*"

In Ambient Calculus it is difficult to count reliably the number of visitors to an ambient. The fix:

$n[\mathit{in}\ m.\ P \mid Q] \mid m[\underline{\mathit{in}}\ m.\ R \mid S]$	$\longrightarrow$	$m[n[P \mid Q] \mid R \mid S]$	(In)
$m[n[\mathit{out}\ m.\ P \mid Q] \mid \underline{\mathit{out}}\ m.\ R \mid S]$	$\longrightarrow$	$n[P \mid Q] \mid m[R \mid S]$	(Out)
$\mathit{open}\ n.\ P \mid n[\underline{\mathit{open}}\ n.Q \mid R]$	$\longrightarrow$	$P \mid Q \mid R$	(Open)
$(m).P \mid \langle M \rangle.Q$	$\longrightarrow$	$P\{m \leftarrow M\} \mid Q$	(Comm)

The Ambient Calculus is recovered by sprinkling $!\underline{\mathit{in}}\ n$, $!\underline{\mathit{out}}\ n$, $!\underline{\mathit{open}}\ n$ appropriately.

Channeled Ambients [Pericas-Geertsen]

Each ambient contains a list of channels c that are used for named communication within the ambient. They are restricted as usual.

$n[D, c;\ c\langle M\rangle.P \mid c(m).Q \mid R]$			(Send)
$\longrightarrow\ \ n[D, c;\ P \mid Q\{m \leftarrow M\} \mid R]$			
$n[D;\ in\ m.\ P \mid Q] \mid m[E;\ R]$	$\longrightarrow$	$m[E;\ n[D;\ P \mid Q] \mid R]$	(In)
$m[E;\ n[D;\ out\ m.\ P \mid Q] \mid R]$	$\longrightarrow$	$n[D;\ P \mid Q] \mid m[E;\ R]$	(Out)
$m[D;\ open\ n.\ P \mid n[E;\ Q] \mid R]$	$\longrightarrow$	$m[D;\ P \mid Q \mid R]$	(Open)

Boxed Ambients [Bugliesi, Castagna, Crafa]

I/O to parents/children is tricky to encode reliably in Ambient Calculus, but is a very natural basic primitive.

Boxed Ambients provide it directly (simplifying Seal):

$n[in\ m.\ P \mid Q] \mid m[R]$	$\longrightarrow m[n[P \mid Q] \mid R]$	(In)
$m[n[out\ m.\ P \mid Q] \mid R]$	$\longrightarrow n[P \mid Q] \mid m[R]$	(Out)
		no (Open)
$(m).P \mid \langle M\rangle.Q$	$\longrightarrow P\{m\leftarrow M\} \mid Q$	(Local)
$(m)^n.P \mid n[\langle M\rangle.Q \mid R]$	$\longrightarrow P\{m\leftarrow M\} \mid n[Q \mid R]$	(Input n)
$\langle M\rangle^n.P \mid n[(m).Q \mid R]$	$\longrightarrow P \mid n[Q\{m\leftarrow M\} \mid R]$	(Output n)
$\langle M\rangle.P \mid n[(m)^{\uparrow}.Q \mid R]$	$\longrightarrow P \mid n[Q\{m\leftarrow M\} \mid R]$	(Input $\uparrow$)
$(m).P \mid n[\langle M\rangle^{\uparrow}.Q \mid R]$	$\longrightarrow P\{m\leftarrow M\} \mid n[Q \mid R]$	(Output $\uparrow$)

Ambjects [Bugliesi, Castagna]

[CG] Ambient Calculus + [AC] Object Calculus =

$$n.a(M).P \mid n[D;\ a(m).Q;\ R] \qquad \text{(Send)}$$

$$\longrightarrow \quad P \mid Q\{m \leftarrow M,\ self \leftarrow n\} \mid n[D;\ a(m).Q;\ R]$$

$$n[D;\ in\ m.\ P \mid Q] \mid m[E;\ R] \longrightarrow m[E;\ n[D;\ P \mid Q] \mid R] \qquad \text{(In)}$$

$$m[E;\ n[D;\ out\ m.\ P \mid Q] \mid R] \longrightarrow n[D;\ P \mid Q] \mid m[E;\ R] \qquad \text{(Out)}$$

$$m[E;\ open\ n.\ P \mid n[D;\ Q] \mid R] \longrightarrow m[E;\ D;\ P \mid Q \mid R] \qquad \text{(Open)}$$

Joinbients [Anonymous]

Ambient Calculus + Join Calculus =

$??? \quad n[D; P]$			(Join)
$n[D; in\ m.\ P \mid Q] \mid m[E; R]$	$\longrightarrow$	$m[E; n[D; P \mid Q] \mid R]$	(In)
$m[E; n[D; out\ m.\ P \mid Q] \mid R]$	$\longrightarrow$	$n[D; P \mid Q] \mid m[E; R]$	(Out)
$m[E; open\ n.\ P \mid n[D; Q]]$	$\longrightarrow$	$m[E; D; P \mid Q]$	(Open)

BioAmbients [Shapiro, Cardelli, et. al.]

Nameless membranes

$[in\ n.\ P \mid Q] \mid [\underline{in}\ n.\ R \mid S]$	$\longrightarrow$	$[[P \mid Q] \mid R \mid S]$	(In)
$[[out\ n.\ P \mid Q] \mid \underline{out}\ n.\ R \mid S]$	$\longrightarrow$	$[P \mid Q] \mid [R \mid S]$	(Out)
$[merge\ n.\ P \mid Q] \mid [\underline{merge}\ n.\ R \mid S]$	$\longrightarrow$	$[P \mid Q \mid R \mid S]$	(Merge)
…			(Comm)

Daring Classification

	Will work fine on a: **LAN** (bounded-delay, integrated management, uniform access)	Will work fine on a: **WAN** (unbounded-delay, federated management, restricted access)
F- (fixed processes and locations)	Calculi (synch/asynch-)π, d-π Infrastructure DOOP Apps File Servers	Calculi π-i, join Infrastructure SOAP, B2C, B2B, P2P Apps Email, Web, Kazaa
M- (mobile processes or locations)	Calculi d-join Soft Infrastructure AGLETS Soft Apps Trusted Applets Hard Infrastructure Wireless Ethernet Hard Apps Work during meetings	Calculi ambients, …, seals Soft Infrastructure Mobile Threads Soft Apps Worms, Agents? Hard Infrastructure Wireless Telephony Hard Apps Mobile B2C, B2B

Conclusions

Studied many encodings, type systems, and equivalences.

- Often building on π-calculus technology.

"Strong mobility" is still a dream, in practice.

- Although many interesting techniques have been proposed, typically in Java.

Ambients suggest new security models.

- Location-based; perhaps more intuitive.
- Analysis of security boundaries.
- But new security issues are also raised.

Ambients are "more" than π.

- Still don't know how to encode ambients in π (vice versa is easy).
- For a generalization of both Ambients and π, see Milner's BiGraphs.