Type-Based Security for *Mobile Computing*

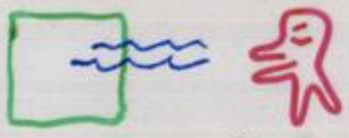
Nobuko Yoshida

Department of Computing
Imperial College London

Type-Based Security for Mobile Computing

- The open characteristic of mobile computing introduces fundamental security issues.
- Example: E-commerce mobile agent

  - Integrity (Access Control) the agent may modify sensitive information

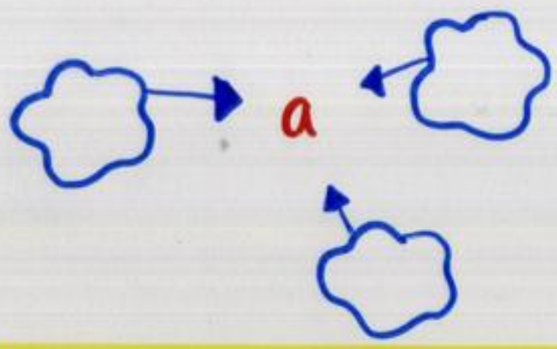


  - Privacy (Secrecy) the agent may transfer credit card numbers to the public channels

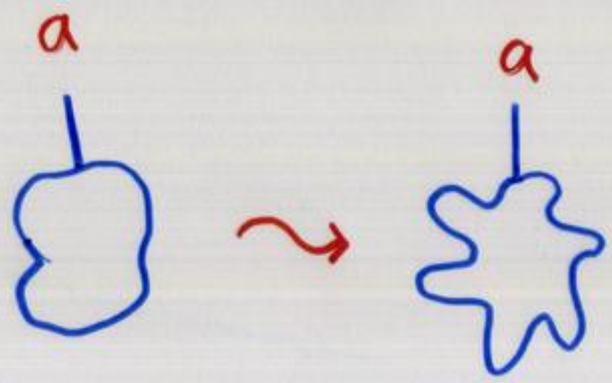
  - Availability (Liveness) the agent may enter into an infinite loop, consuming unbounded local resources.

The π-Calculus [1989–]

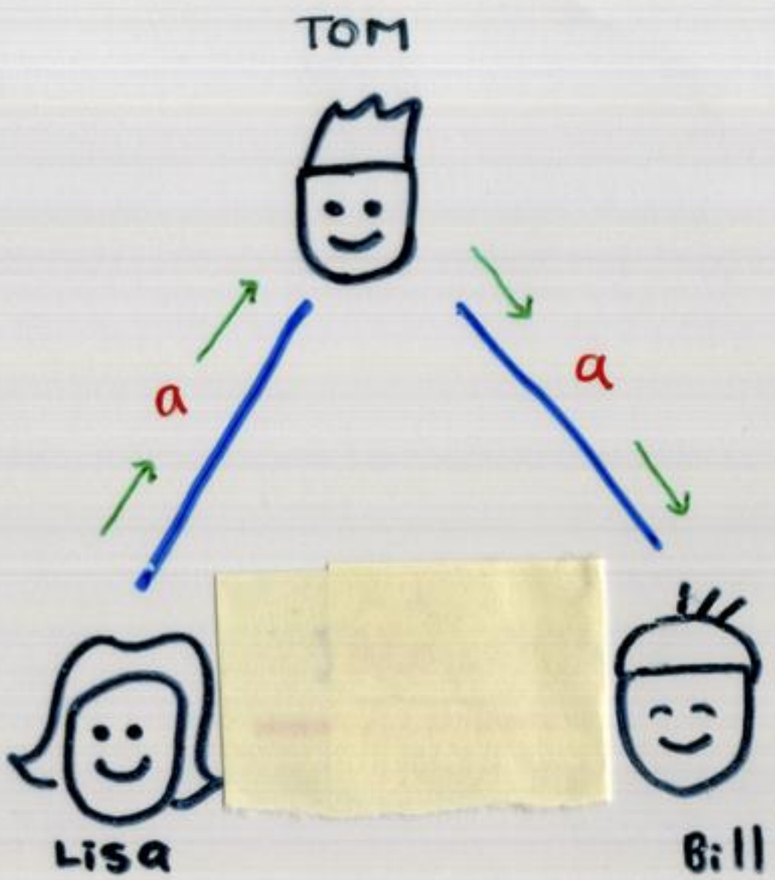
Shared Point



Identity of Agent

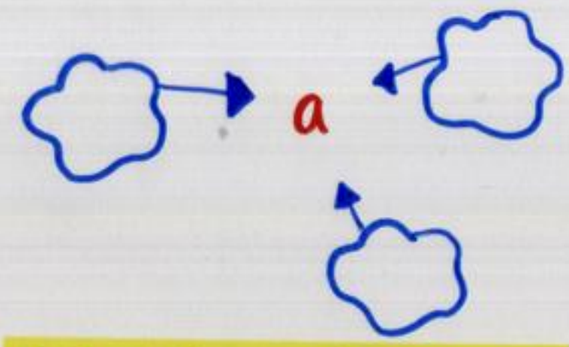


Name Passing as Pointer Passing

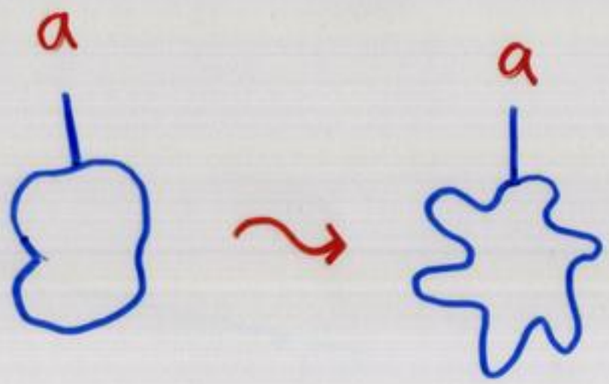


The π-Calculus [1989–]

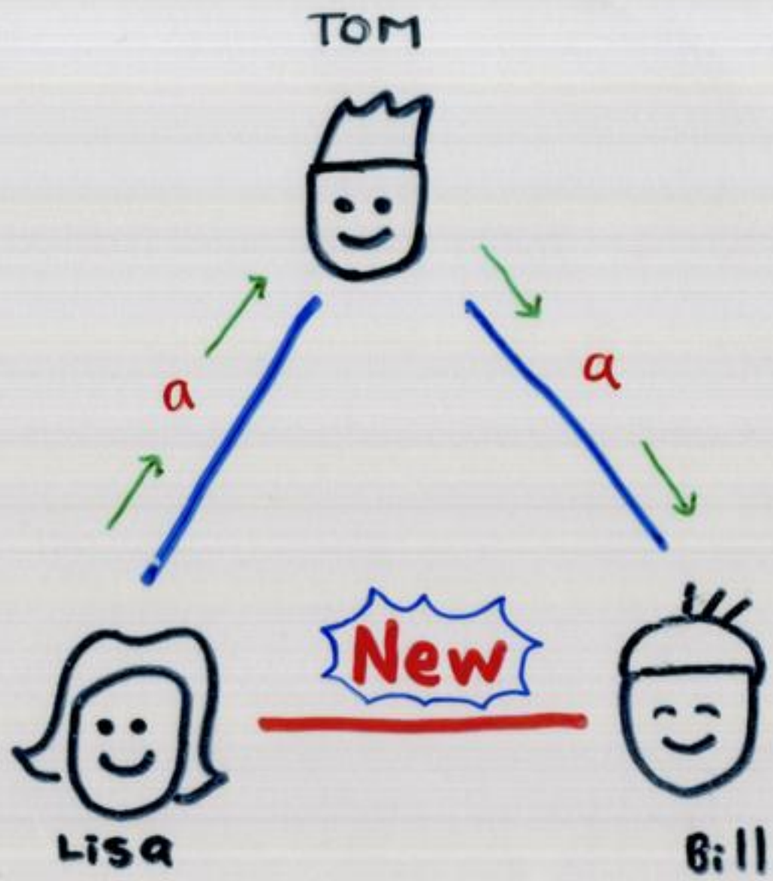
Shared Point



Identity of Agent



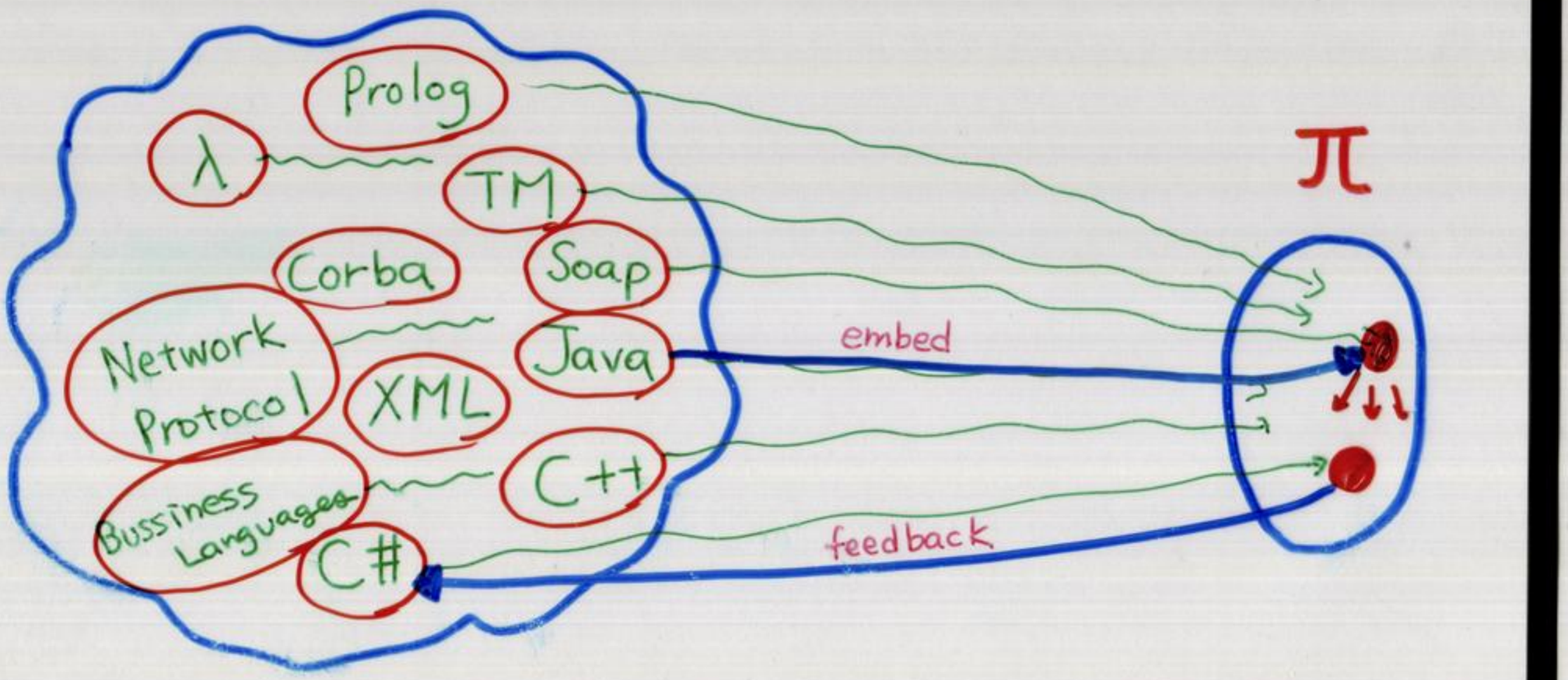
Name Passing as Pointer Passing



The π-Calculus

object message

$$a(x).P \mid \bar{a}v \quad \rightarrow \quad P\{v/x\}$$



Full Abstraction by Types

[TLCA01]
[LICS01]
[FoSSaCS02]
[FoSSaCS03]
[CW04]
⋮

Higher-Order π-Calculus

[Sangiorgi 93]

CML, Facile, LLinda, ...

λ_v $(\lambda x.Q)V \rightarrow Q[V/x]$

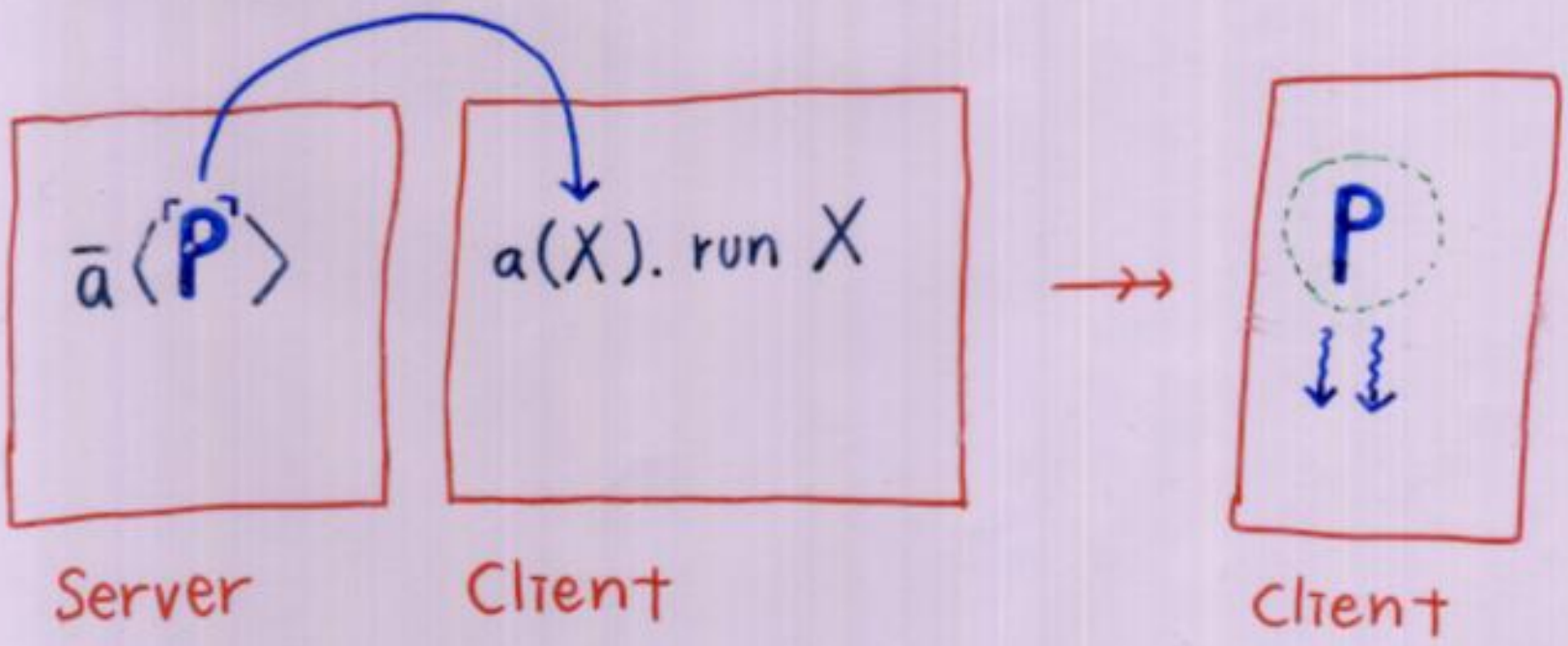
$\lambda\pi_v$ $\bar{a}\langle\ulcorner P \urcorner\rangle \mid a(X).\,run\ X$

Code

$\rightarrow$ $run\ \ulcorner P \urcorner \rightarrow P$

where $\underset{thunk}{\ulcorner P \urcorner} = \lambda().\,P$

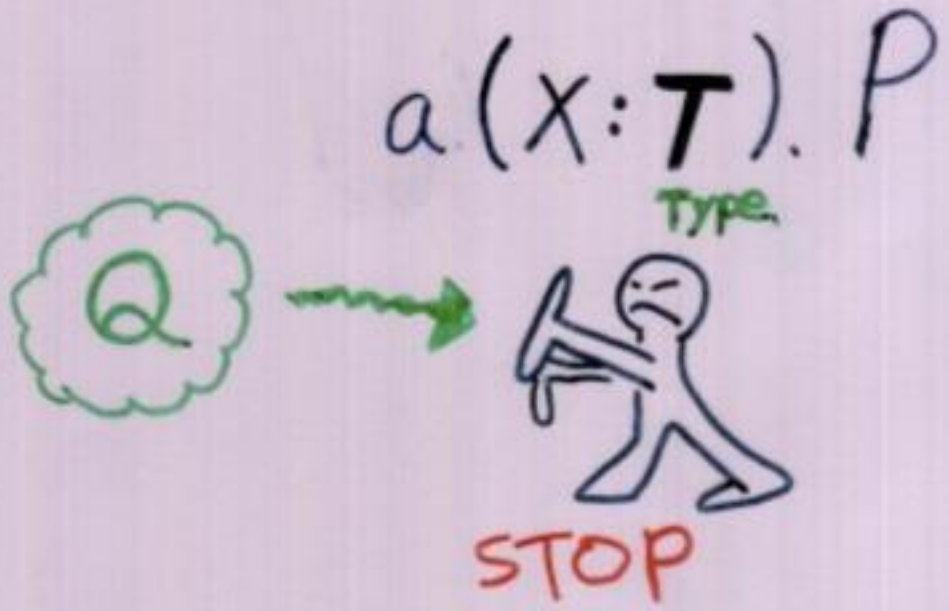
$run = \lambda X.\,X()$



Aims of Types / Typechecking

- Using Types to control the effects of Mobile Code / Processes

- Host refuses to execute incoming code unless it conforms to predetermined access policy



History of Types in HOπ/π

HOπ

93 **HOπ** [Sangiorgi PhD]

98 **Fine-Grained Types**
[Yoshida and Hennessy]

00 **Other Mobility Types**
- Seal Calculus
- M- Calculus
- Safe-Boxed Ambient

03 **Channel Dependency Existential Types**

⇒ Integration with Linearity

π

89 **π-Calculus**

92 Polyadic π

93 IO-Subtyping
Linear Typings
Polymorphism
Causality-based Typings

01 **Fully Abstraction**
- PCF [TCAL 01]
- $\lambda\to+\times$ [LICS 01]
- System F [FoSSacs 01]
- Control [CW04]

02 **Secure Info Flow**
[ESOP 01, FoSSaCs 02
POPL 02]

⇒ Application
Secure Info
Role-Based Access Control

Problem

$$c(X:\ulcorner proc \urcorner).\ run\ X$$

where $\ulcorner proc \urcorner = unit \to proc$

Any Process is welcome

P $\underline{a}(X).\ \underline{\bar{b}}\langle X\rangle$

Q $\underline{b}(X).\ \underline{\bar{a}}\langle X\rangle$

R $\underline{d}\langle string\rangle$

Assigning Types to Processes

[Yoshida and Hennessy 2000]

$$\Gamma \vdash P : [\Delta]$$

channel environment = INTERFACE

Example

P $\quad \Gamma \vdash \underline{a}(x).\,\overline{\underline{b}}\langle x\rangle : [a:(T)^{I},\ b:(T)^{O}]$

Q $\quad \Gamma \vdash \underline{b}(x).\,\overline{\underline{a}}\langle x\rangle : [a:(T)^{O},\ b:(T)^{I}]$

R $\quad \Gamma \vdash d\langle string\rangle : [\,d:(string)^{O}]$

$$c(X:\ulcorner a:(T)^{I},\ b:(T)^{O}\urcorner).\ run\ X$$

P	O
Q	X
R	X
0	O

History of Types in HOπ/π

HOπ

93 **HOπ** [Sangiorgi PhD]

98 **Fine-Grained Types**
[Yoshida and Hennessy]

00 **Other Mobility Types**
- Seal Calculus
- M- Calculus
- Safe-Boxed Ambient

03 **Channel Dependency Existential Types**

⇒ Integration with Linearity

π

89 **π-Calculus**

92 Polyadic π

93 IO-Subtyping
Linear Typings
Polymorphism
Causality-based Typings

01 **Fully Abstraction**
- PCF [TCAL 01]
- $\lambda\to+\times$ [LICS 01]
- System F [FoSSacs 01]
- Control [CW04]

02 **Secure Info Flow**
[ESOP 01, FoSSaCs 02
POPL 02]

⇒ Application
Secure Info
Role-Based Access Control

Main Theorems

Subject Reduction

$$\Gamma \vdash P : \Sigma \;,\; P \twoheadrightarrow P' \Rightarrow \Gamma \vdash P' : \Sigma$$

Type Safety

$$\Gamma \vdash P : [\Delta] \Rightarrow P \xrightarrow{\Gamma,\not[\Delta]} err$$

where $P \xrightarrow{\Gamma,\not[\Delta]} err$ means

P can use at most resources in Δ

Consequence:

$$a(X:\ulcorner\Delta\urcorner).P \mid \bar{a}\langle\ulcorner R\urcorner\rangle \xrightarrow{\Gamma,\pi} err$$

if $\Gamma \not\vdash R : [\Delta]$

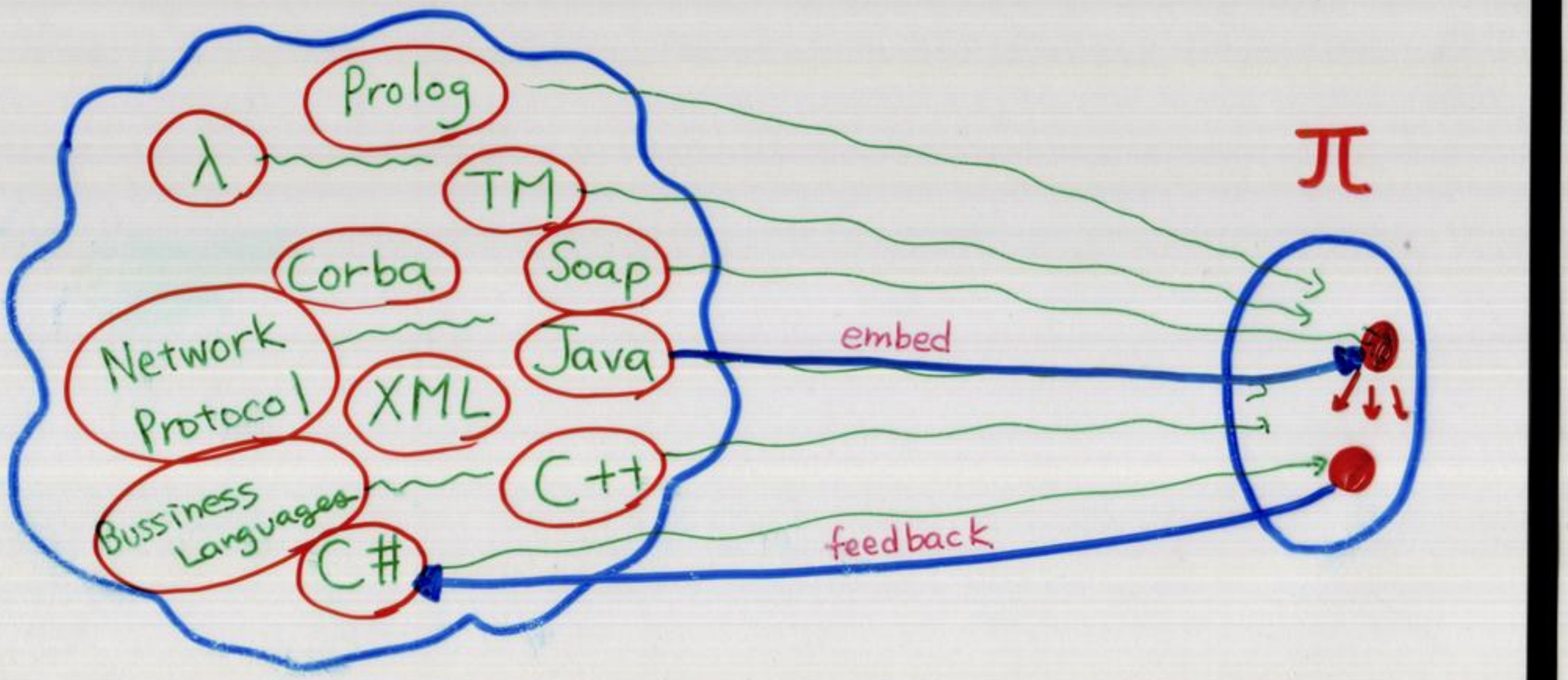
The Project Aim

- to develop a general and uniform framework for type systems of mobile programs using process calculi.
  - (Base Language) $\Longrightarrow$ Liveness and Integrity
    An integrated framework using typed π
  - (Mobility) $\Longrightarrow$ Access Control
    A fine-grained typing system for HOπ
  - (Application) $\Longrightarrow$ Language Design Discipline
    A secure distributed multi-threaded Java and a proof of its correctness
- Novelty directly impacting language/system design from the initial stage

The π-Calculus

object message

$$a(x).P \mid \bar{a}v \quad \rightarrow \quad P\{v/x\}$$



Full Abstraction by Types

[TLCA01]
[LICS01]
[FoSSaCS02]
[FoSSaCS03]
[CW04]
⋮

Conclusion

- Timeliness and Impacts

  W3C Choreography Working Group www.w3c.org

- Many themes for ambitious RAs and PhDs, involving both theory and software design and implementation

- Contact

  - Nobuko Yoshida yoshida@doc.ic.ac.uk

    Home Page: www.doc.ic.ac.uk/~yoshida

  - Kohei Honda kohei@dcs.qmul.ac.uk

    Home Page: www.dcs.qmul.ac.uk/~kohei