programming in

Ambience

gearing up for dynamic adaptation to context

Pleiad Research Meetings

Sebastián González

24 June 2009

Département d'ingénierie informatique
École Polytechnique de Louvain
Université catholique de Louvain

Programming desktops and servers

Programming for Ambient Intelligence

Programming for Ambient Intelligence

Programming for Ambient Intelligence

Programming for Ambient Intelligence

Programming for Ambient Intelligence

Programming for Ambient Intelligence

Programming for Ambient Intelligence

Programming for Ambient Intelligence

... is programming with context in mind

New hardware phenomena unveil new opportunities for delivering specific services according to the physical and logical environment of use

Examples

peer service
take advantage of room projector for presentation

location semantics
disable phone ringtone in quiet places

internal state
decrease playback quality when battery power is low

user task
show parking spots and gas stations while driving

environmental conditions
give more detailed indications when visibility is low

Running Example

call reception behaviour

Running Example

call reception behaviour

context behaviour

Running Example

call reception behaviour

context	behaviour
default	ringtone

Running Example

call reception behaviour

context

quiet

behaviour

vibration

Running Example

call reception behaviour

context	behaviour
off-hook	call waiting signal

Running Example

call reception behaviour

context	behaviour
unavailable	forward call

Running Example

call reception behaviour

context

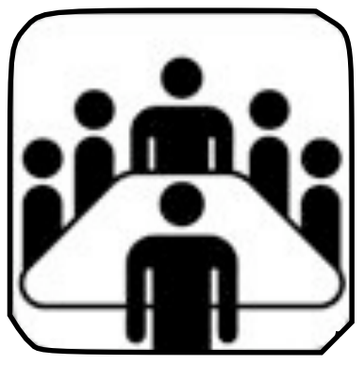
behaviour

off-hook + quiet

Running Example

call reception behaviour

context			behaviour
off-hook	+	quiet	call waiting signal



Contemporary Solution

conditional statements

```
class phone {
    method receive ( call ) {
        if ( phone.isOffHook( ) )
            play( phone.callWaitingSignal( ), 2 );
        else if ( phone.environment( ).acoustics( ).isQuiet( ) )
            phone.vibrate( 5 );
        else if ( phone.user( ).isUnavailable( ) )
            forwardCall( call, phone.forwardNumber( ) );
        else
            play( phone.ringTone( ), 10 );
    }
```

Contemporary Solution

conditional statements

```
class phone {
    method receive ( call ) {
        if ( phone.isOffHook( ) )
            play( phone.callWaitingSignal( ), 2 );
        else if ( phone.environment( ).acoustics( ).isQuiet( ) )
            phone.vibrate( 5 );
        else if ( phone.user( ).isUnavailable( ) )
            forwardCall( call, phone.forwardNumber( ) );
        else
            play( phone.ringTone( ), 10 );
}
```

Tangled
Scattered
Fixed

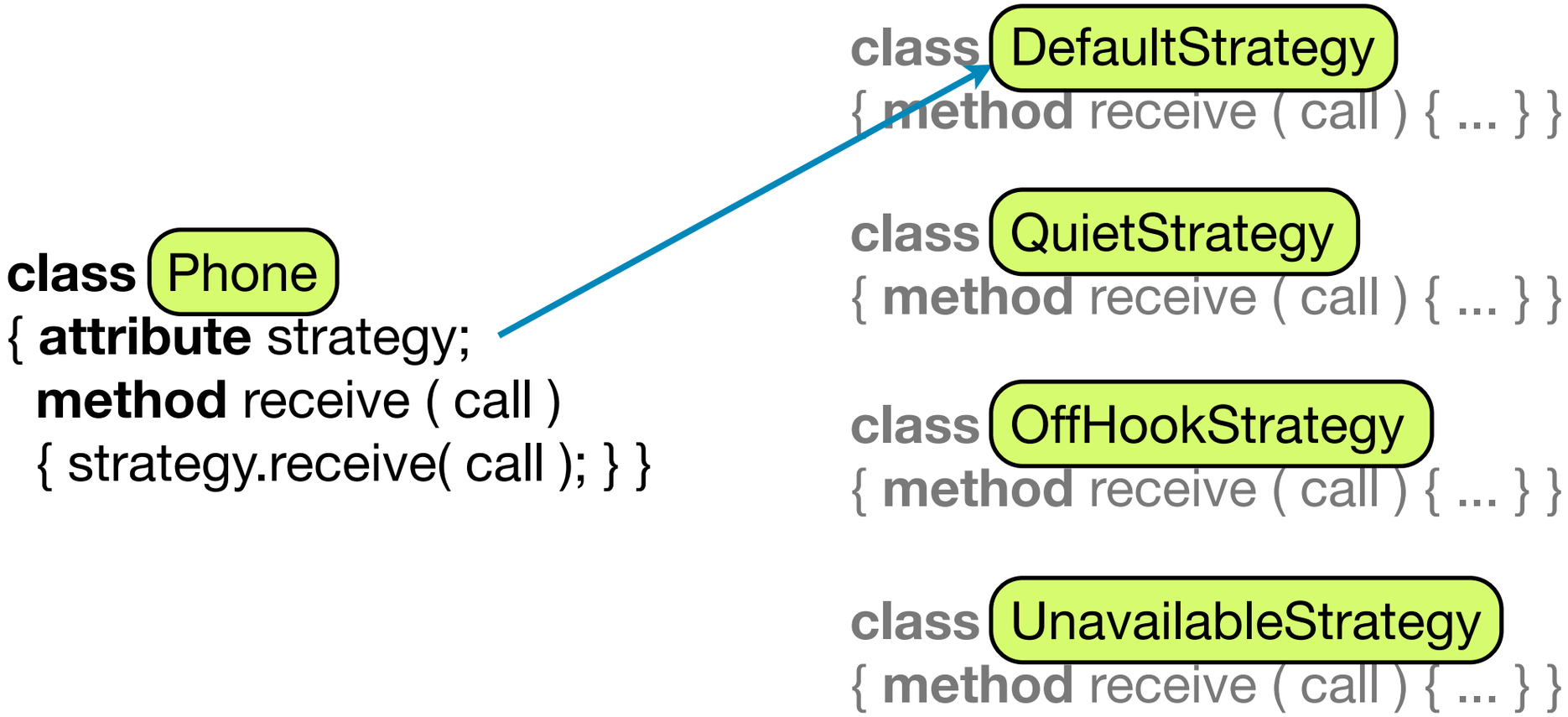
Contemporary Solution

special software architecture

```
class  Phone
{ attribute strategy;
  method receive ( call )
  { strategy.receive( call ); } }
```

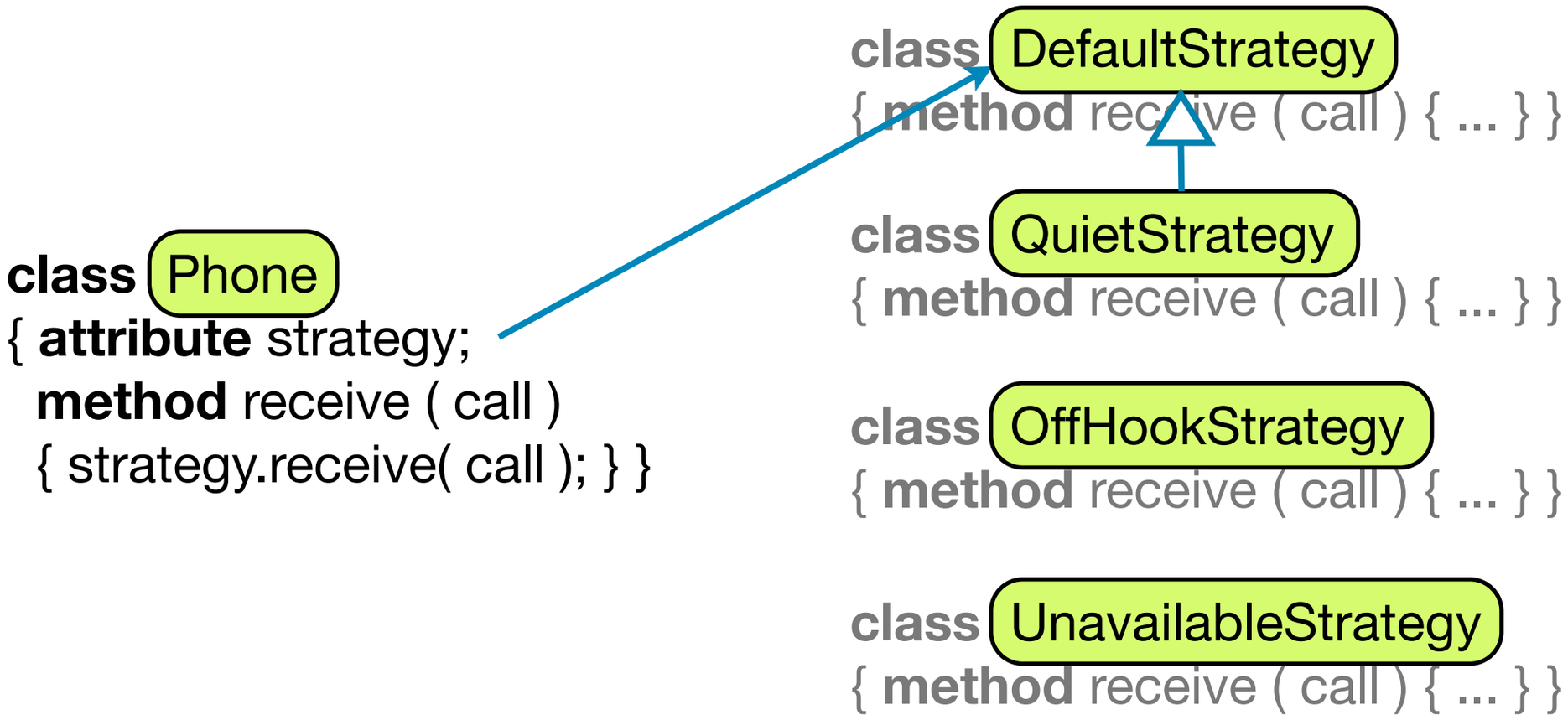
Contemporary Solution

special software architecture



Contemporary Solution

special software architecture



Contemporary Solution

special software architecture

class Phone
{ **attribute** strategy;
method receive (call)
{ strategy.receive(call); } }

class DefaultStrategy
{ **method** receive (call) { ... } }

class QuietStrategy
{ **method** receive (call) { ... } }

class OffHookStrategy
{ **method** receive (call) { ... } }

class UnavailableStrategy
{ **method** receive (call) { ... } }

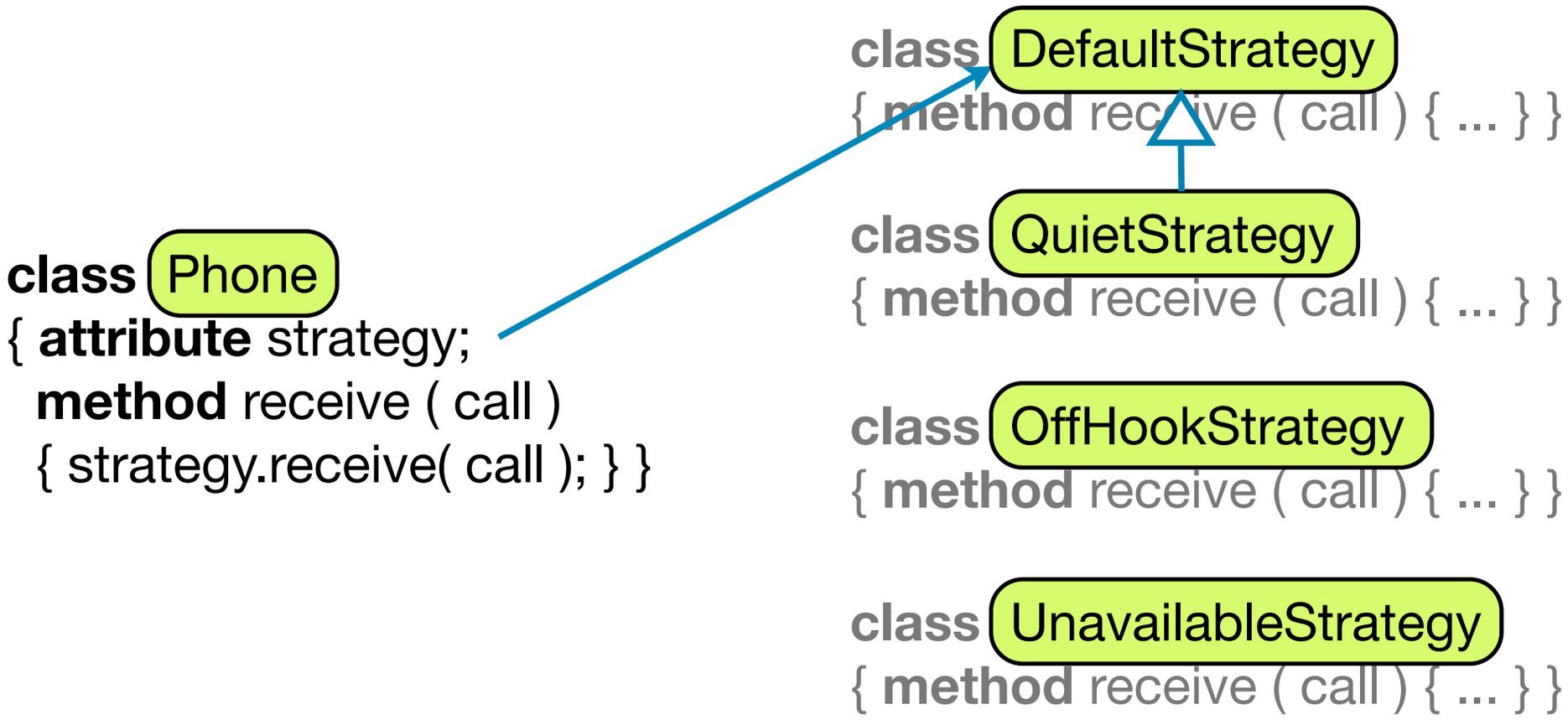
Infrastructural burden

Anticipated adaptation points

in Ambient Intelligence

dynamic adaptation is commonplace

rather than an occasional need



Infrastructural burden

Anticipated adaptation points

From programming in isolation...

Programming languages do not feature dedicated abstractions to deal with contexts and corresponding behavioural adaptations at run time

... to programming in Ambience

Develop programming abstractions to permit the natural expression of adaptable behaviour according to changing contexts

Solution

The **Ambient Object System (AmOS)**

a new computation model aimed at dynamic behaviour adaptation

The **Ambience programming language**

a syntactic skin for the underlying ambient object system

1. dynamic behaviour adaptation
2. straightforward application logic, without extrinsic context adaptation concerns
3. simplified application logic thanks to adaptation to intrinsic contexts
4. non-intrusive expression of adaptations
5. straightforward software architectures

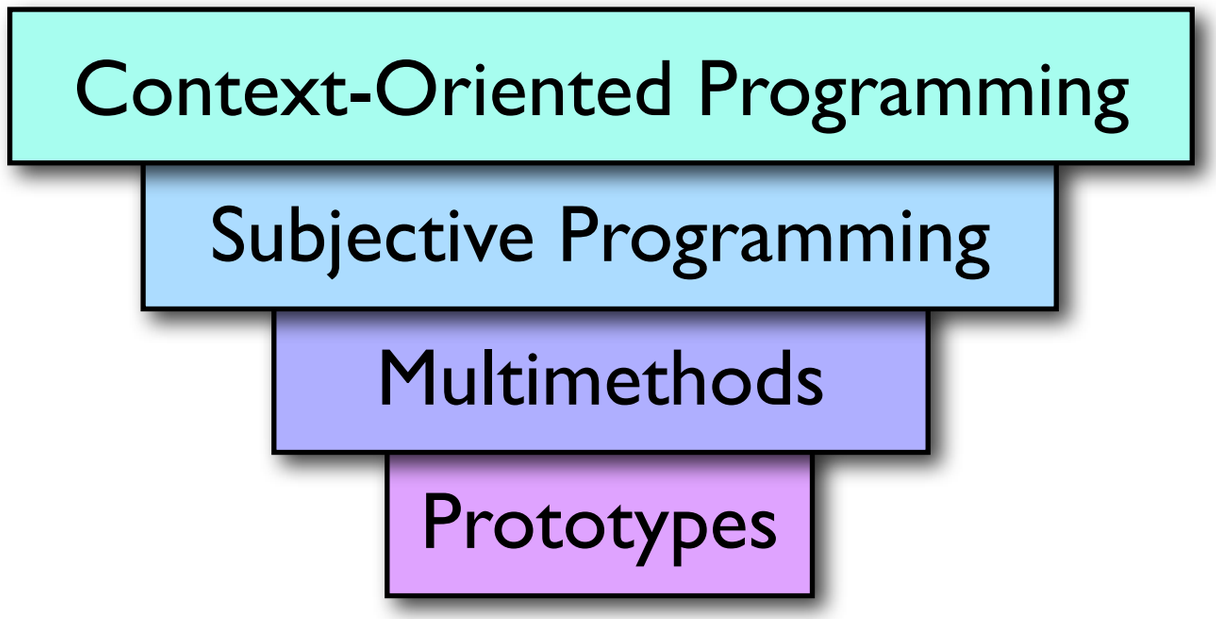
Solution

The **Ambient Object System (AmOS)**

a new computation model aimed at dynamic behaviour adaptation

The **Ambience programming language**

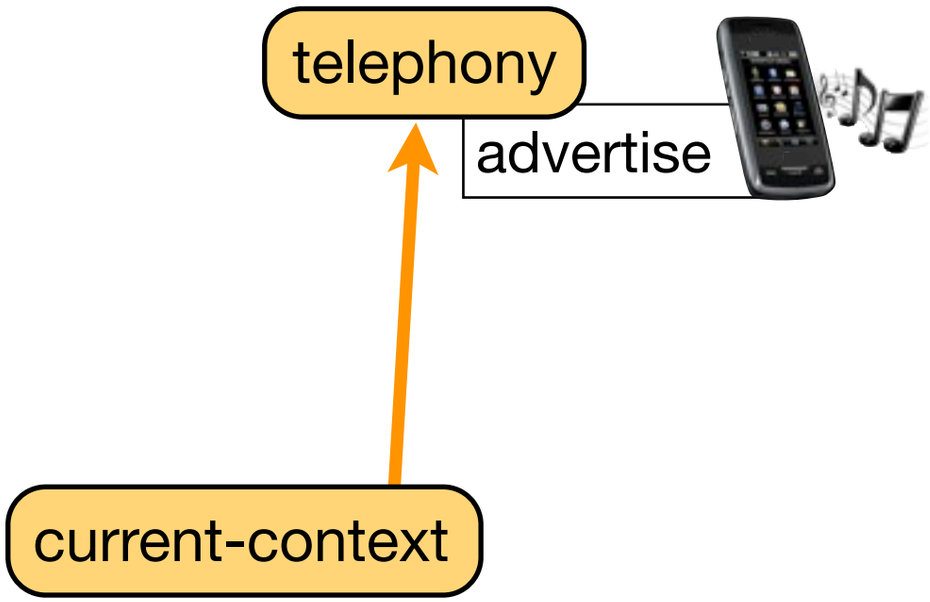
a syntactic skin for the underlying ambient object system



Context Reification

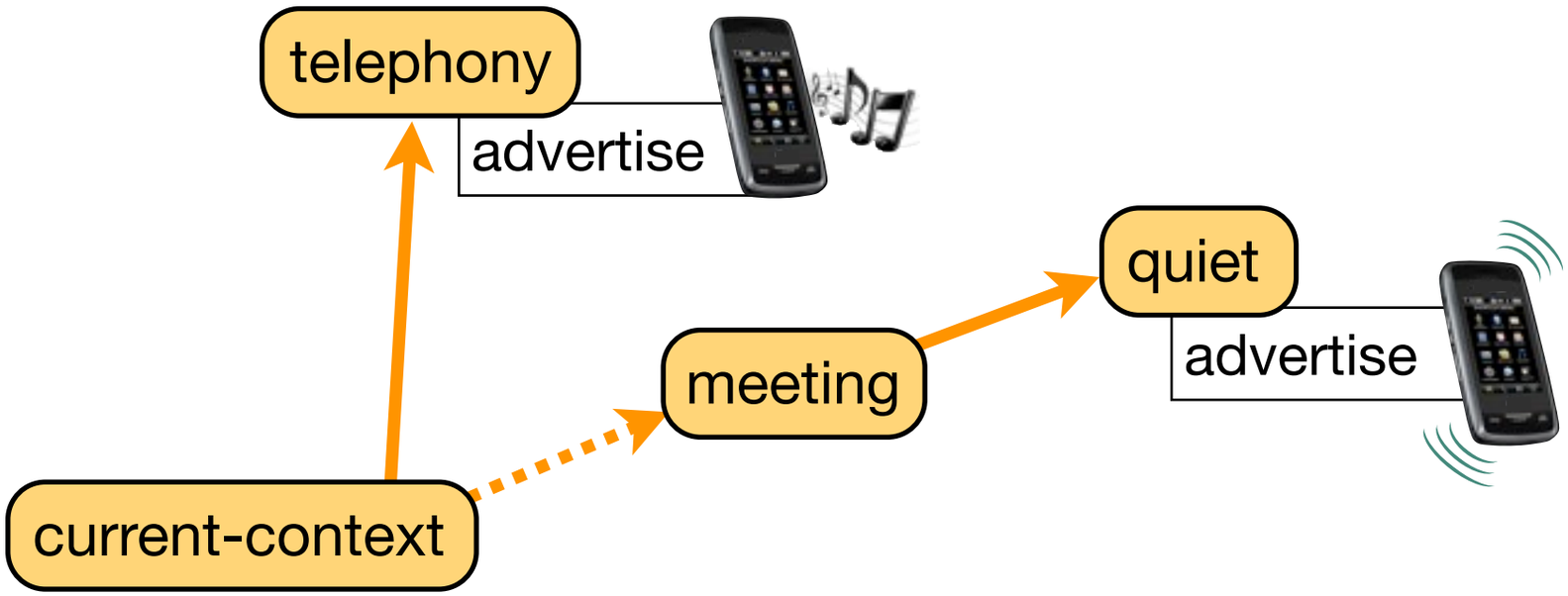
First-class context

Context Reification



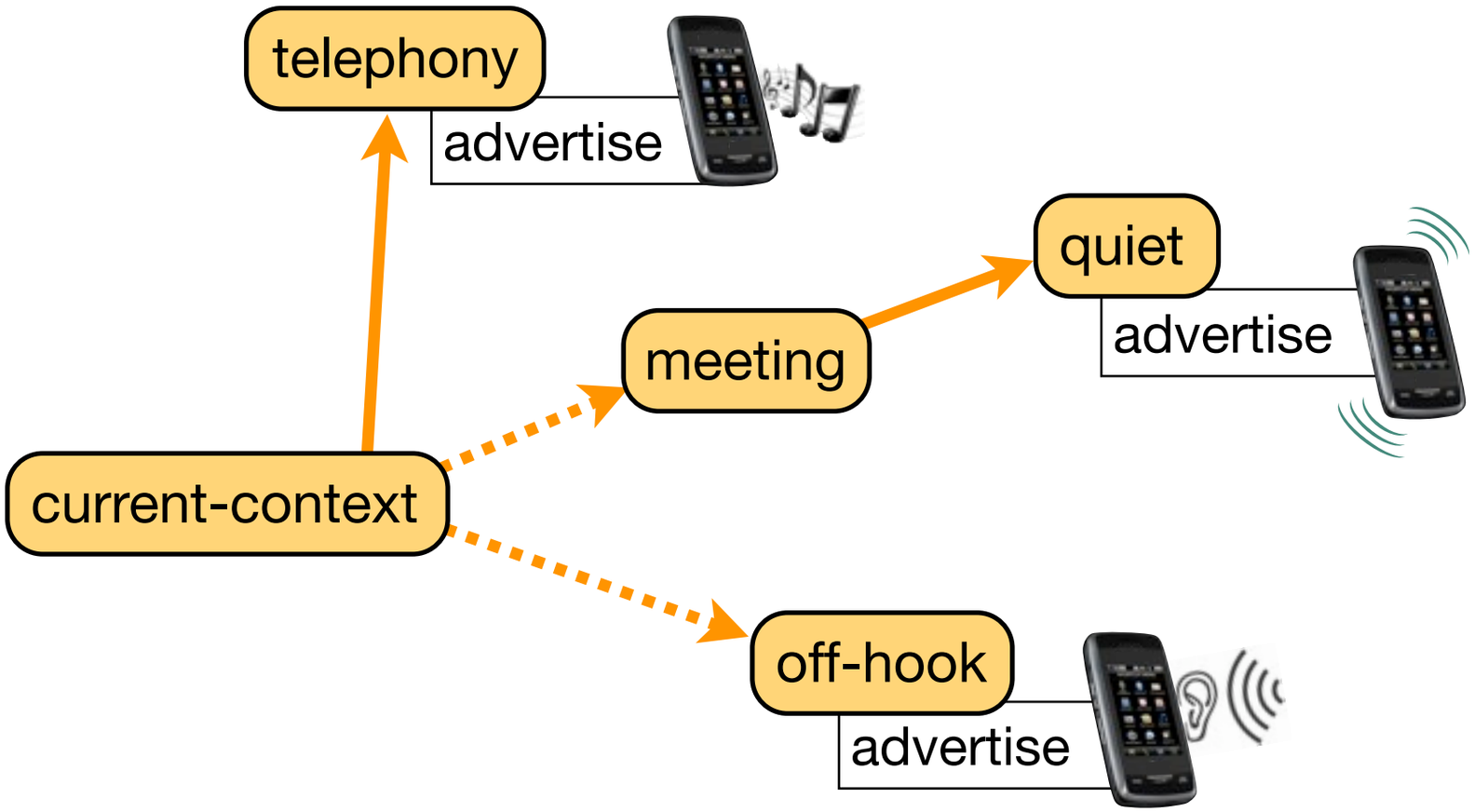
First-class context

Context Reification



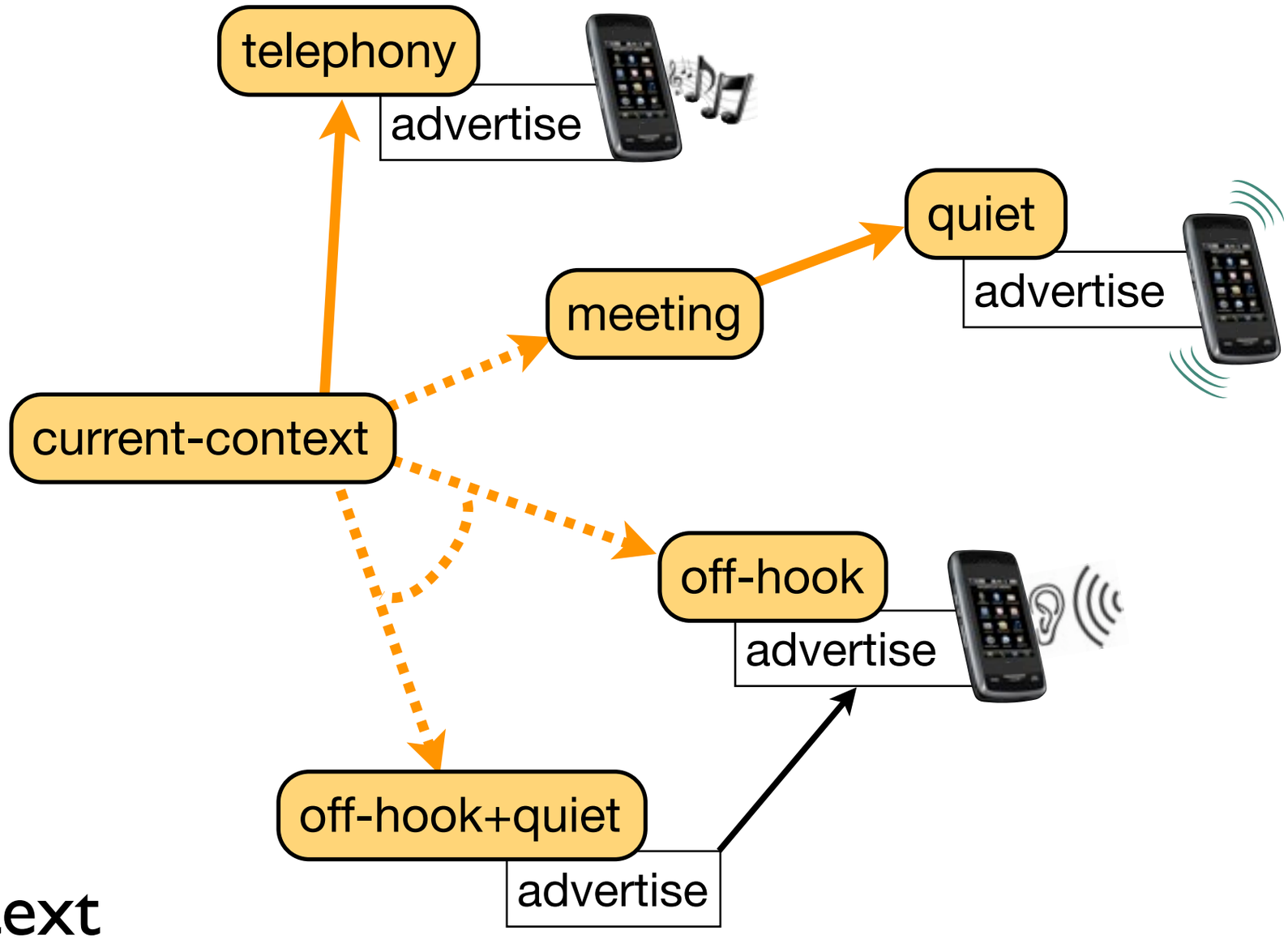
First-class context

Context Reification



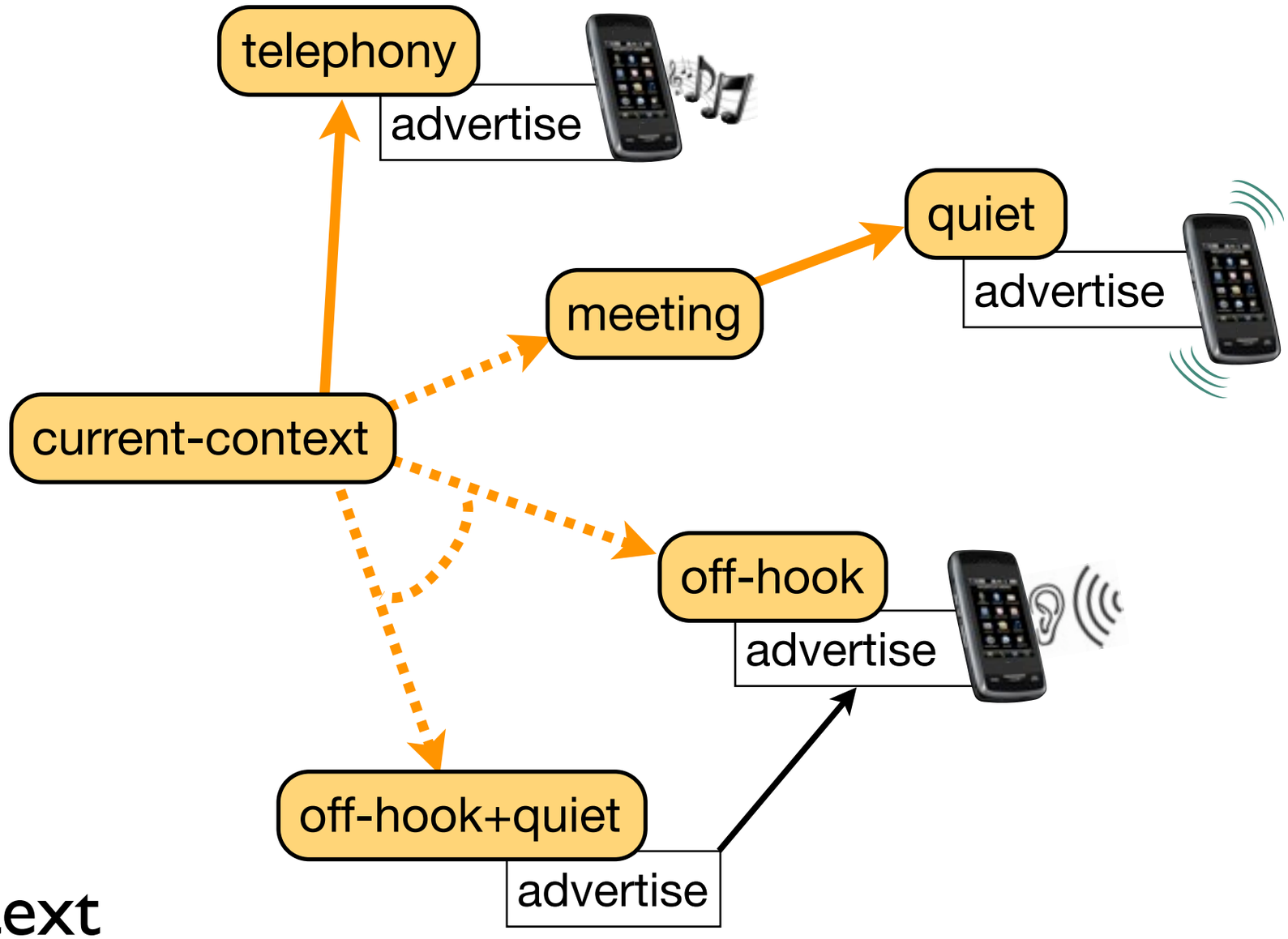
First-class context

Context Reification



First-class context

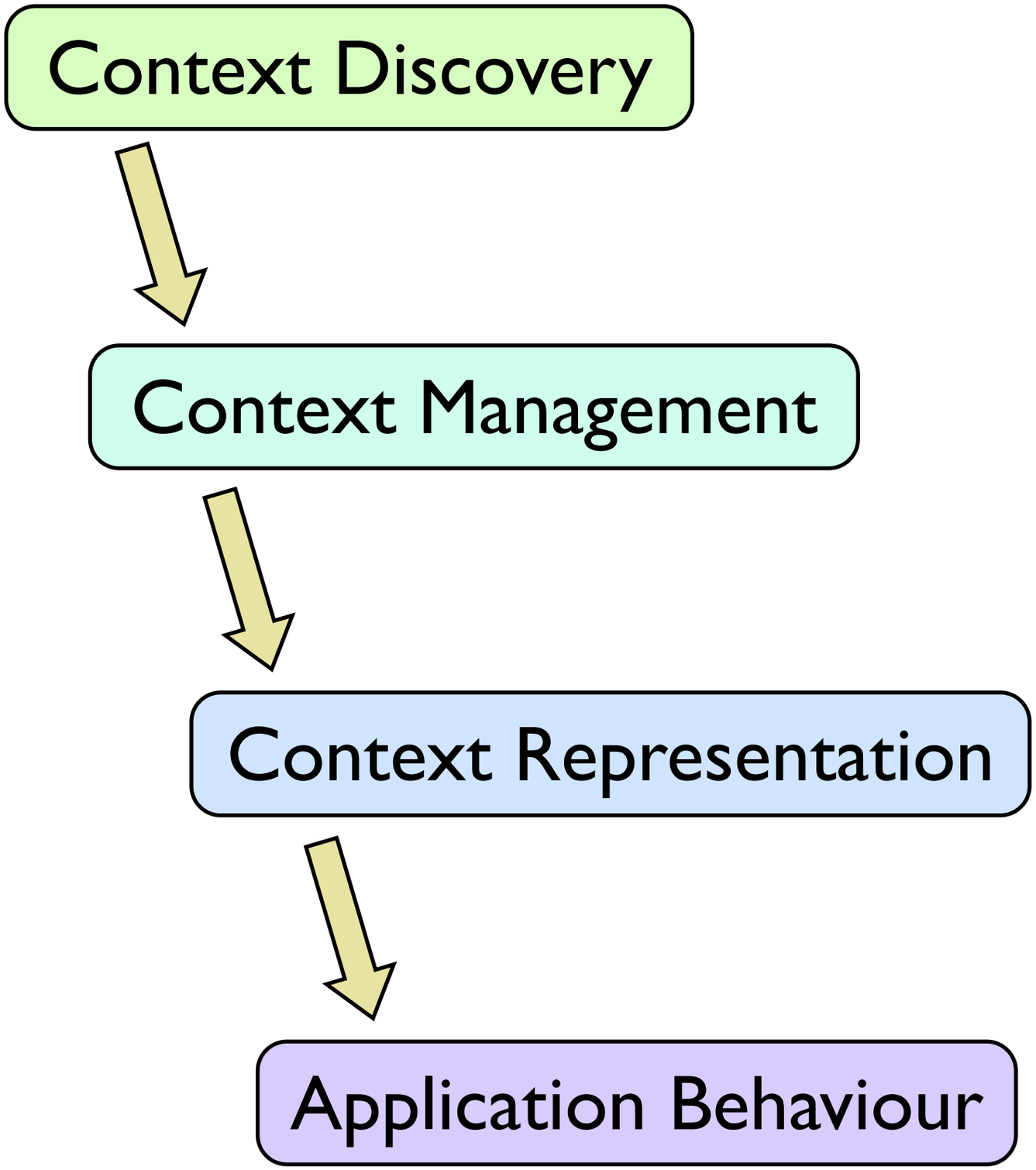
Context Reification



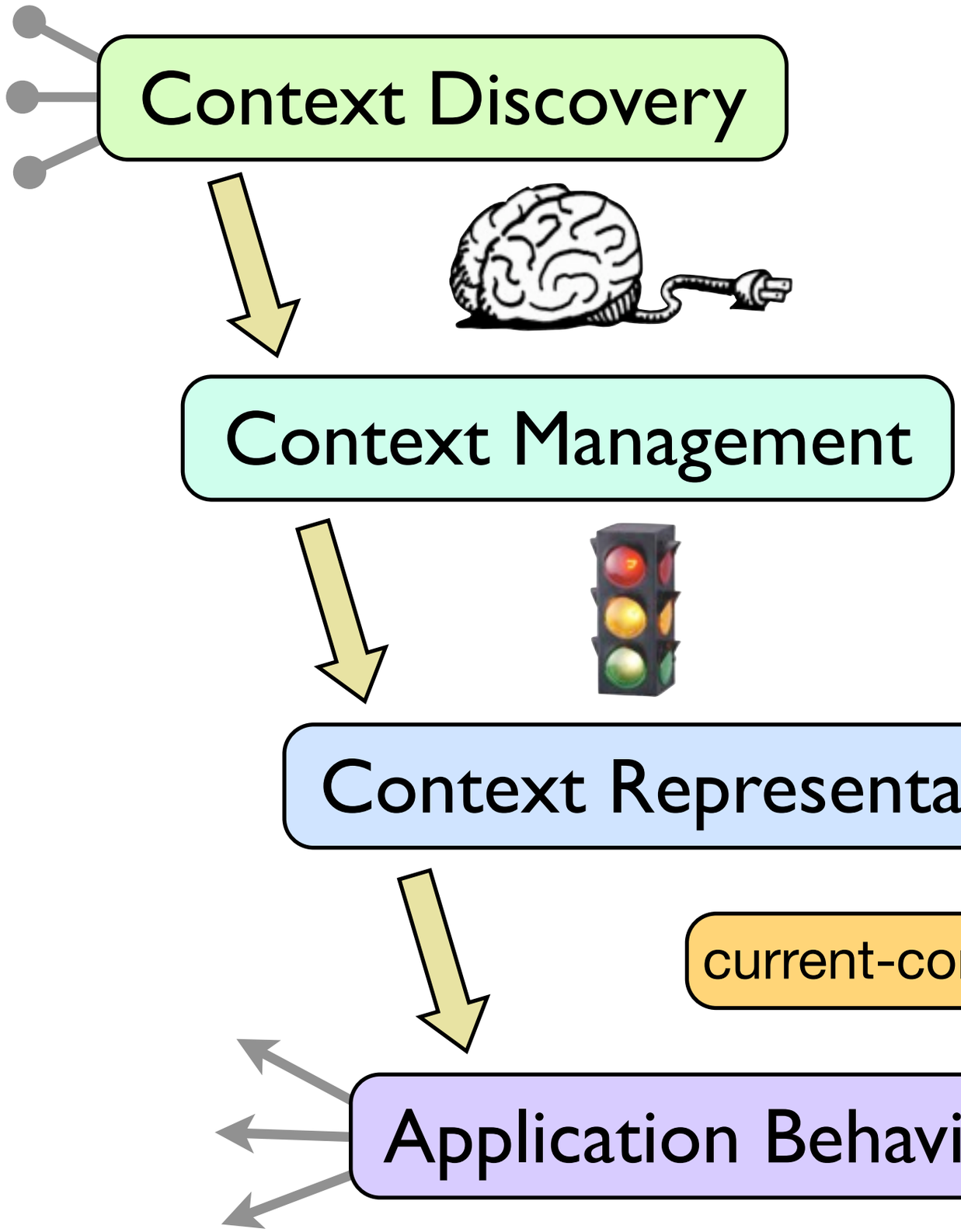
First-class context

- partly managed by user
- partly managed by system

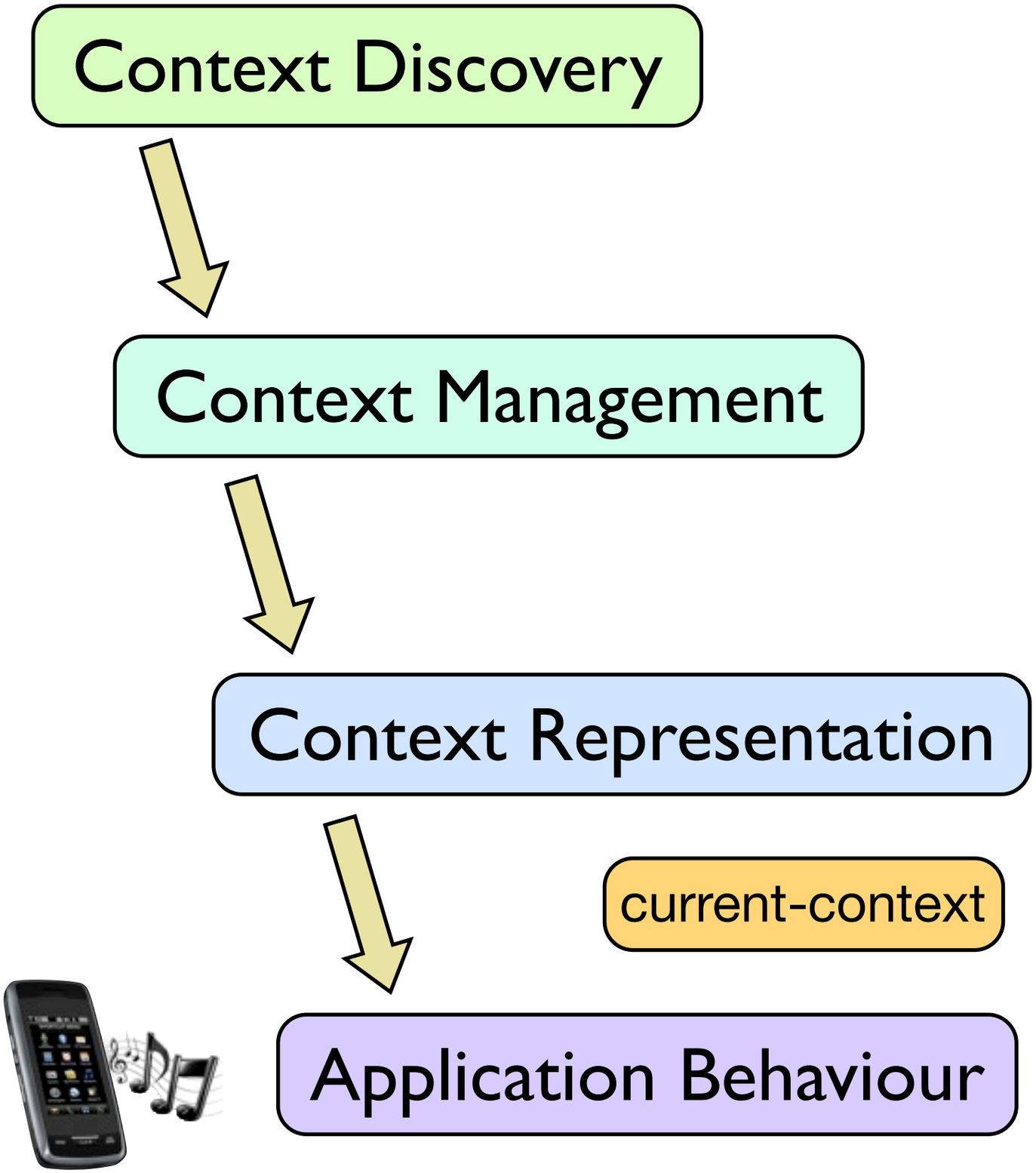
Context-Aware System Architecture



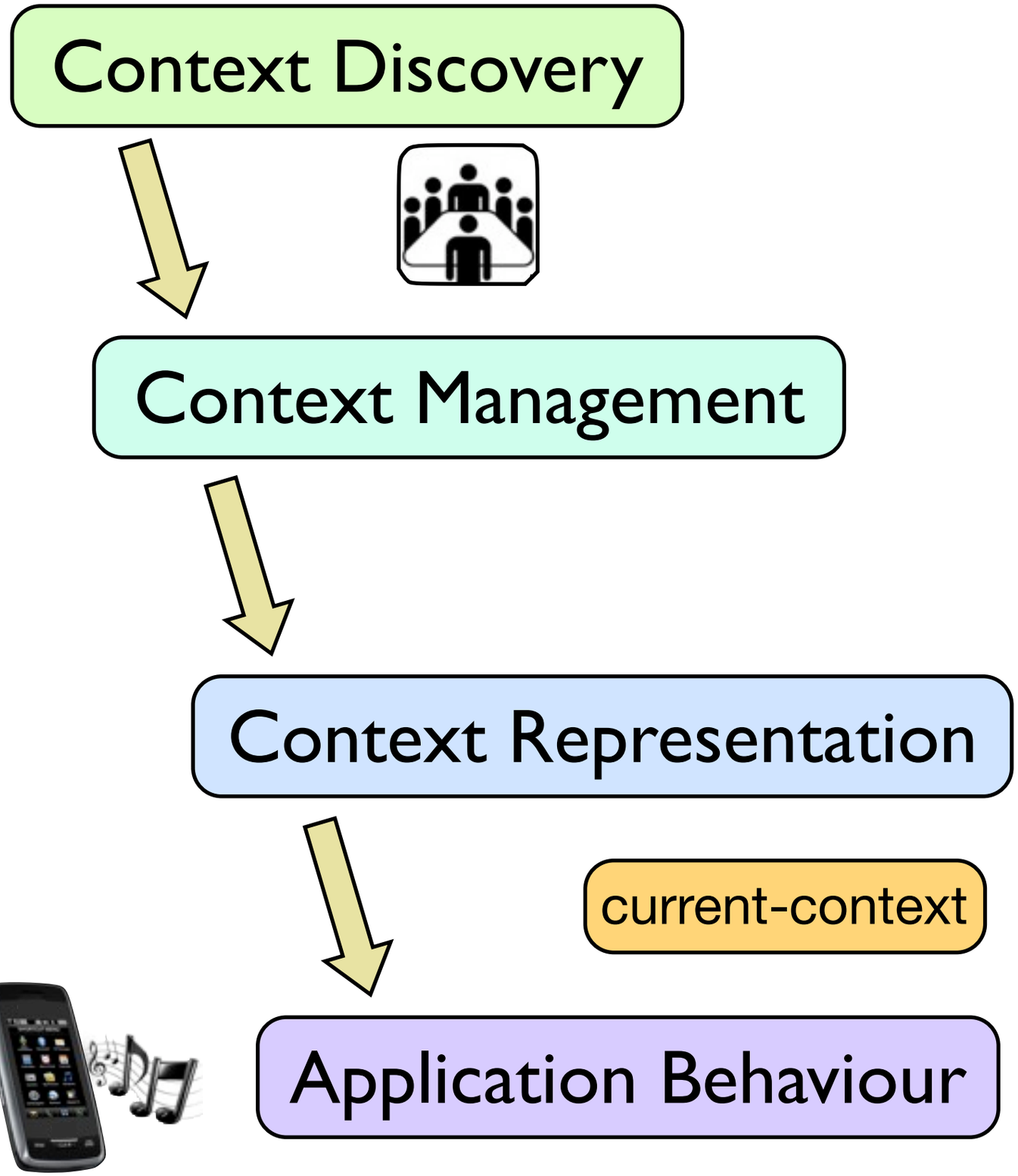
Context-Aware System Architecture



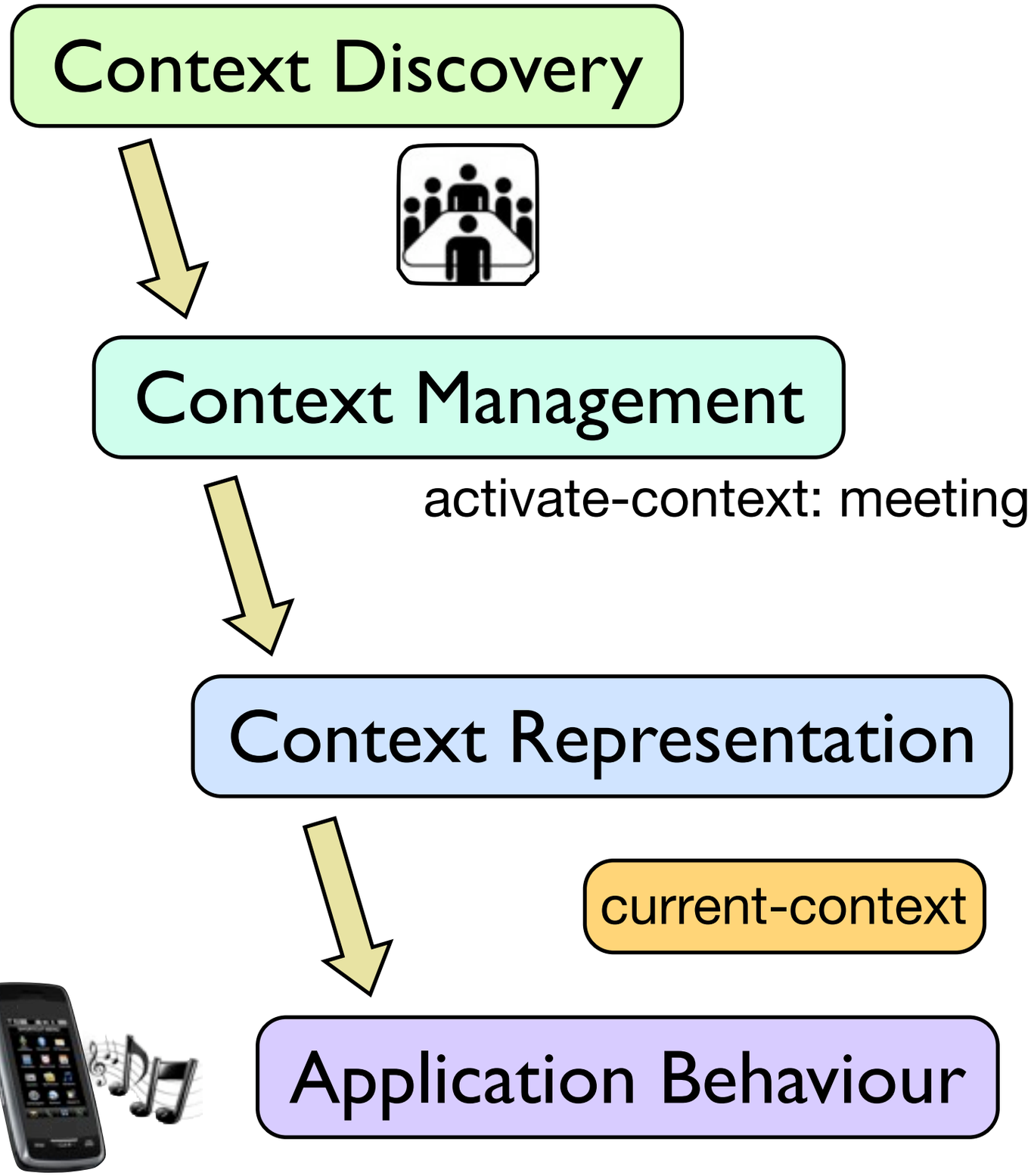
Context-Aware System Architecture



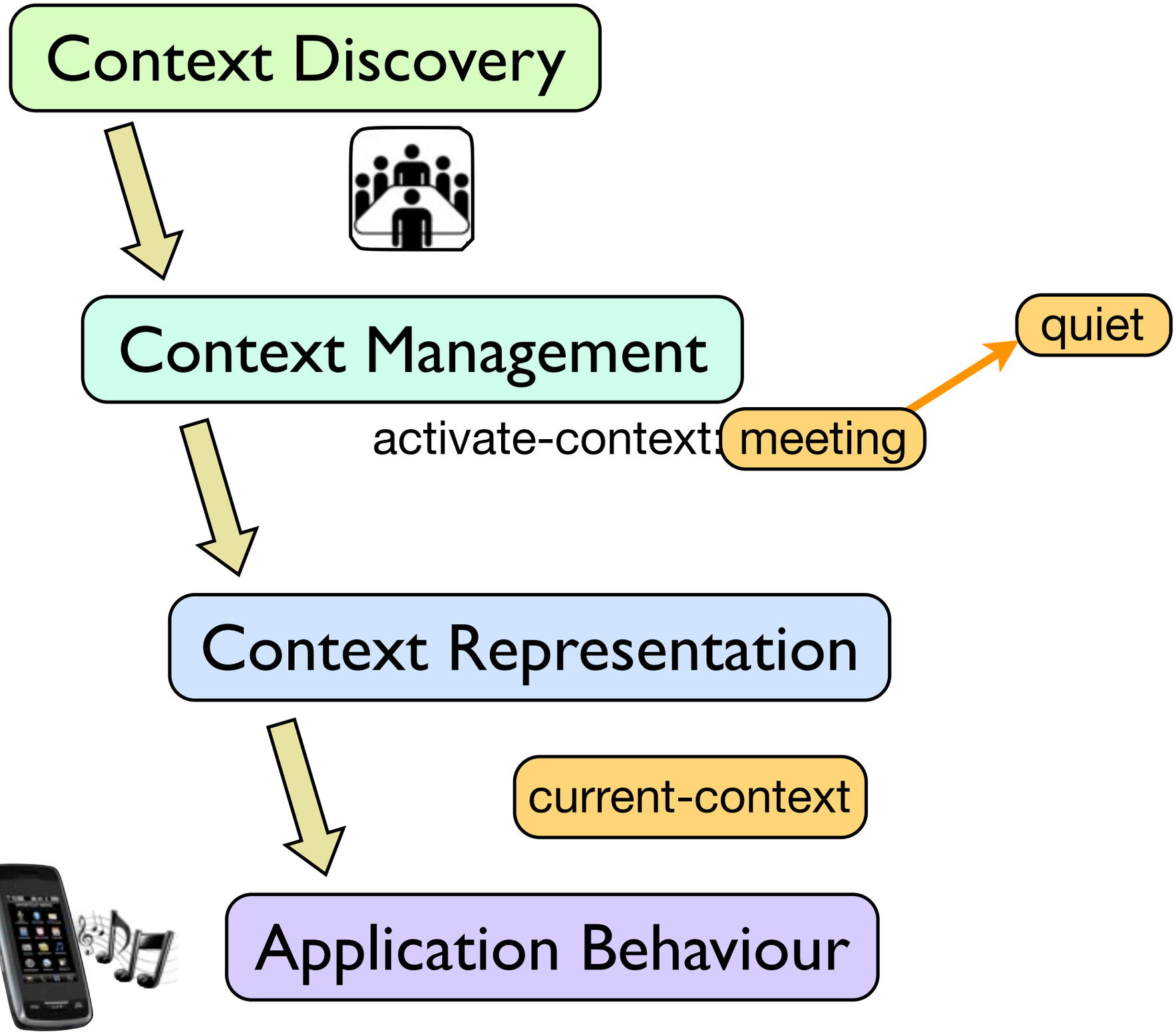
Context-Aware System Architecture



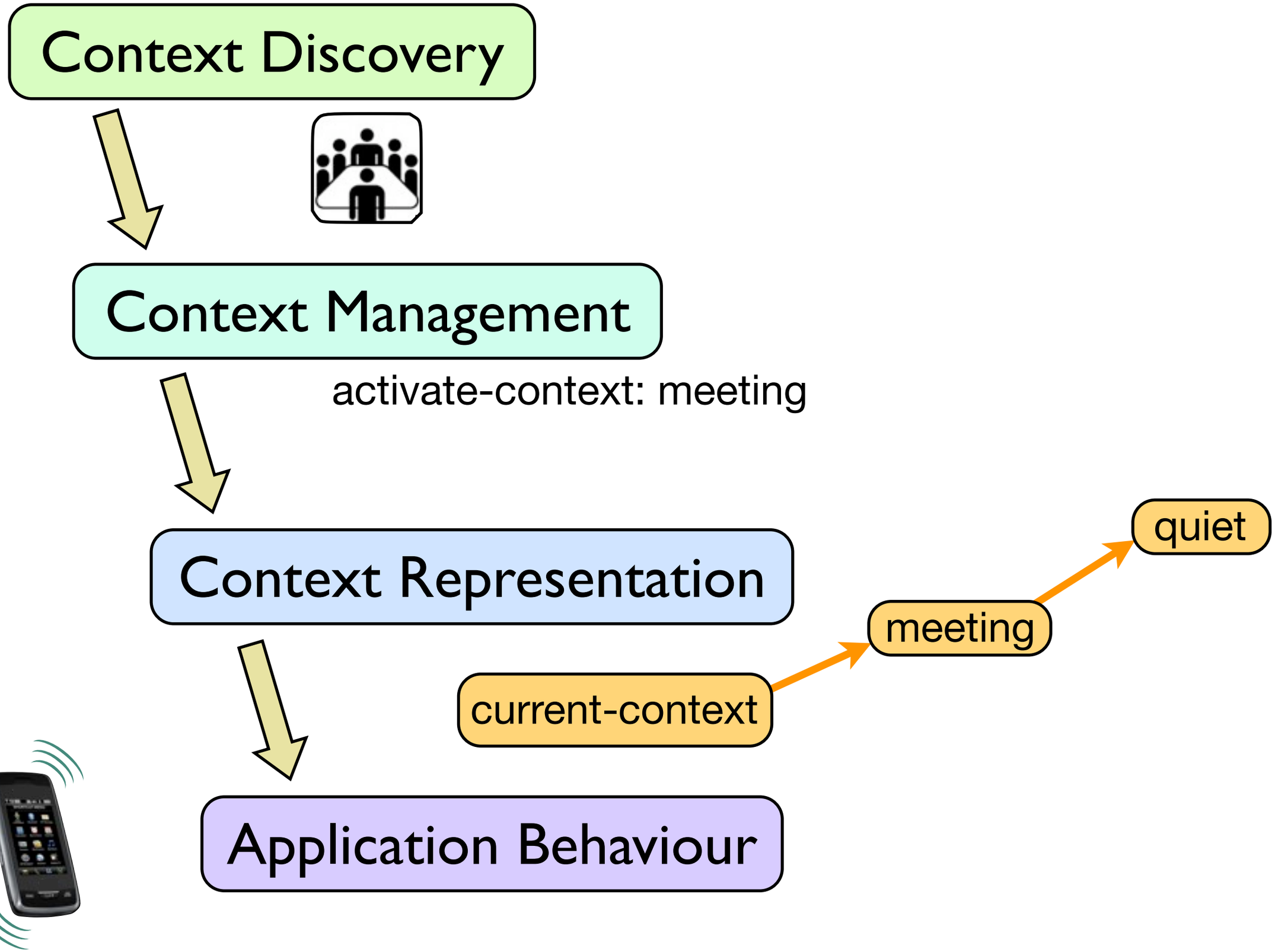
Context-Aware System Architecture



Context-Aware System Architecture



Context-Aware System Architecture



Context-Aware System Architecture

Context Discovery

Context Management

activate-context: meeting
activate-context: handsfree

Context Representation

quiet

meeting

current-context

handsfree

Application Behaviour

Context-Aware System Architecture

Context Discovery

Context Management

activate-context: meeting
activate-context: handsfree
activate-context: off-hook

Context Representation

current-context

- meeting
- quiet
- handsfree
- off-hook

Application Behaviour

Context-Aware System Architecture

Context Discovery

Context Management

activate-context: meeting
activate-context: handsfree
activate-context: off-hook

Context Representation

Application Behaviour

current-context

meeting

quiet

handsfree

off-hook

dynamic
behaviour
adaptation

Context-Aware System Architecture

Context Discovery

Context Management

activate-context: meeting
activate-context: handsfree
activate-context: off-hook

Context Representation

current-context

- meeting
- quiet
- handsfree
- off-hook

Application Behaviour

Running Example in Ambience

Core concepts and syntax

messages

```
receive:  alices-call  on:  bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

receive: alices-call on: bobs-phone

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

```
receive:  alices-call  on:  bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

```
receive:  alices-call  on:  bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

```
receive: alices-call on: bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

```
receive:  alices-call  on:  bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
 [ advertise: call on: phone.
   enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

receive: alices-call on: bobs-phone

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

```
receive:  alices-call  on:  bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

```
receive:  alices-call  on:  bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

```
receive:  alices-call  on:  bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

```
receive:  alices-call  on:  bobs-phone
```

methods
(symmetric)

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

objects

messages

```
receive: alices-call on: bobs-phone
```

methods
(symmetric)

```
receive: call phone-call on: phone mobile-phone
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

receive: alices-call on: bobs-phone

objects

delegation
(multiple, dynamic)

methods
(symmetric)

```
receive: call phone-call on: phone mobile-phone
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

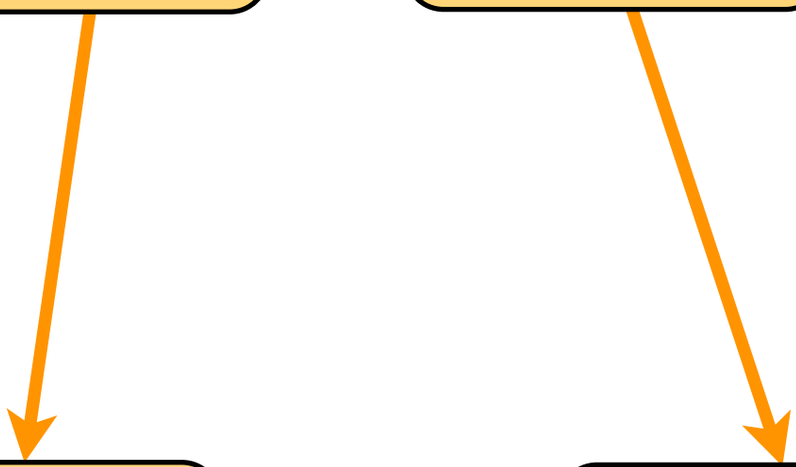
messages

receive: alices-call on: bobs-phone

objects

delegation
(multiple, dynamic)

ambiguities



receive: call phone-call on: phone mobile-phone
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls]

methods
(symmetric)

Running Example in Ambience

Core concepts and syntax

messages

receive: alices-call on: bobs-phone

objects

delegation
(multiple, dynamic)

ambiguities | C3*

methods
(symmetric)

```
receive: call phone-call on: phone mobile-phone
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Core concepts and syntax

messages

receive: alices-call on: bobs-phone

objects

delegation
(multiple, dynamic)

ambiguities | C3*

methods
(symmetric)

receive: call phone-call on: phone mobile-phone
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls]

multiple dispatch

Running Example in Ambience

Core concepts and syntax

messages

receive: alices-call on: bobs-phone

objects

delegation
(multiple, dynamic)

ambiguities | C3*

methods
(symmetric)

receive: call phone-call on: phone mobile-phone
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls]

multiple dispatch

ambiguities

Running Example in Ambience

Core concepts and syntax

messages

left precedence right

receive: alices-call on: bobs-phone

objects

delegation
(multiple, dynamic)

ambiguities | C3*

methods
(symmetric)

```
receive: call phone-call on: phone mobile-phone
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

multiple dispatch

ambiguities

Running Example in Ambience

Core concepts and syntax

messages

left precedence right

receive: alices-call on: bobs-phone

objects

delegation
(multiple, dynamic)

ambiguities | C3*

methods
(symmetric)

receive: call phone-call on: phone mobile-phone
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls]

(asymmetric) multiple dispatch

ambiguities | arg. precedence

Smalltalk | Self | CLOS | Dylan | Cecil | Slate

Running Example in Ambience

Core concepts and syntax

messages

left precedence right

receive: alices-call on: bobs-phone

objects

delegation
(multiple, dynamic)

ambiguities | C3*

methods
(symmetric)

receive: call phone-call on: phone mobile-phone
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls]

(asymmetric) multiple dispatch

ambiguities | arg. precedence

Running Example in Ambience

Call Reception Behaviour

receive: alices-call on: bobs-phone

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]
```

Running Example in Ambience

Call Reception Behaviour

receive: alices-call on: bobs-phone

receive: call (phone-call) on: phone (mobile-phone)
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls]

advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds]

Running Example in Ambience

Call Reception Behaviour

receive: alices-call on: bobs-phone

application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ]
```

Running Example in Ambience

Call Reception Behaviour

receive: alices-call on: bobs-phone

how do we express adaptations of base application logic to context?

application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ]

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ]
```

Implicit Context Specialisation

receive: alices-call on: bobs-phone

advertise: call (phone-call) on: phone (mobile-phone)
[activate: phone vibrator during: 10 seconds]

advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds]

Implicit Context Specialisation

receive: alices-call on: bobs-phone

in-context: quiet do:
[

advertise: call (phone-call) on: phone (mobile-phone)
[activate: phone vibrator during: 10 seconds]

]

advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds]

Implicit Context Specialisation

implicit explicit

current-activation receive: alices-call on: bobs-phone

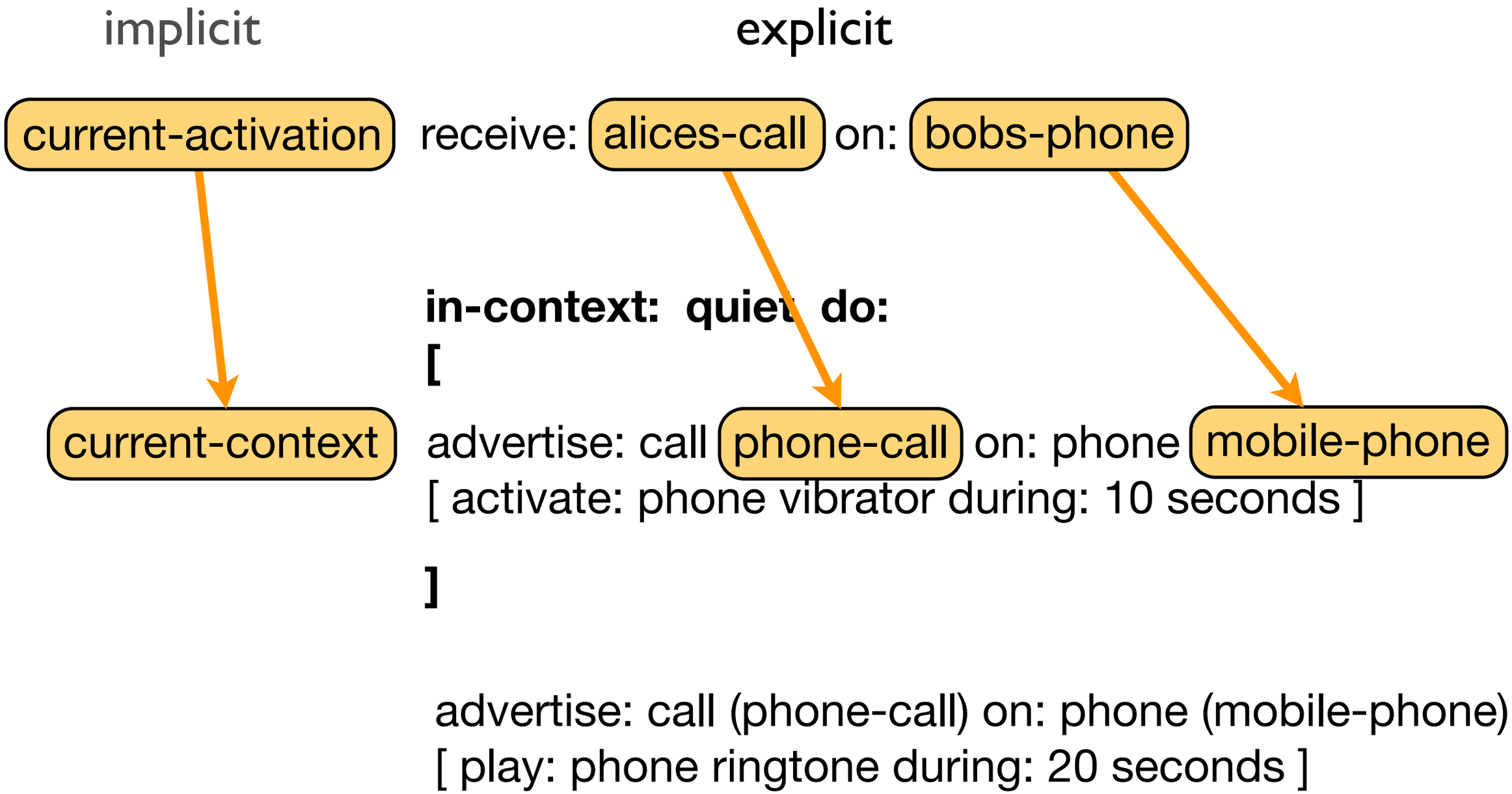
in-context: quiet do:
[

(current-context) advertise: call (phone-call) on: phone (mobile-phone)
[activate: phone vibrator during: 10 seconds]

]

advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds]

Implicit Context Specialisation



Implicit Context Specialisation

receive: alices-call on: bobs-phone

in-context: quiet do:
[

quiet advertise: call phone-call on: phone mobile-phone
[activate: phone vibrator during: 10 seconds]

]

advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds]

Us | ContextL

Implicit Context Specialisation

```
receive:  alices-call  on:  bobs-phone

in-context:  quiet  do:
[

advertise: call (phone-call) on: phone (mobile-phone)
[ activate: phone vibrator during: 10 seconds ]

]


 advertise: call (phone-call) on: phone (mobile-phone)
 [ play: phone ringtone during: 20 seconds ]
```

Context-Specific Behaviour

application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ].
```

adaptation logic

```
in-context:  quiet  do:
[ advertise: call (phone-call) on: phone (mobile-phone)
  [ activate: phone vibrator during: 10 seconds ] ]
```

Context-Specific Behaviour

2 straightforward application logic

receive: call (phone-call) on: phone (mobile-phone)
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls].

advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds].

adaptation logic

in-context: quiet do:
[advertise: call (phone-call) on: phone (mobile-phone)
  [activate: phone vibrator during: 10 seconds] **]**

Context-Specific Behaviour

2 straightforward application logic

receive: call (phone-call) on: phone (mobile-phone)
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls].

advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds].

4 non-intrusive adaptation logic

in-context: quiet do:
[advertise: call (phone-call) on: phone (mobile-phone)
  [activate: phone vibrator during: 10 seconds] **]**

Context-Specific Behaviour

2 straightforward application logic

receive: call (phone-call) on: phone (mobile-phone)
[advertise: call on: phone.
  enqueue: call in: phone incoming-calls].

advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds].

4 non-intrusive adaptation logic

in-context: quiet do:
[advertise: call (phone-call) on: phone (mobile-phone)
  [activate: phone vibrator during: 10 seconds] **]**

extrinsic to telephony

Context-Specific Behaviour

application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ].
```

adaptation logic

```
in-context:  quiet  do:
[ advertise: call (phone-call) on: phone (mobile-phone)
  [ activate: phone vibrator during: 10 seconds ] ]
```

Basic Telephony Behaviour

application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ if: phone is-off-hook
  then: [ play: phone call-waiting-signal during: 3 seconds ]
  else: [ play: phone ringtone during: 20 seconds ] ].
```

Basic Telephony Behaviour

application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ if: phone is-off-hook
  then: [ play: phone call-waiting-signal during: 3 seconds ]
  else: [ play: phone ringtone during: 20 seconds ] ].
```

Fixed

Basic Telephony Behaviour

application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ].
```

application logic

```
in-context:  off-hook  do:
[ advertise: call (phone-call) on: phone (mobile-phone)
  [ play: phone call-waiting-signal during: 3 seconds ] ]
```

Basic Telephony Behaviour

2 straightforward application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ].
```

application logic

```
in-context:  off-hook  do:
[ advertise: call (phone-call) on: phone (mobile-phone)
  [ play: phone call-waiting-signal during: 3 seconds ] ]
```

Basic Telephony Behaviour

2 straightforward application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ].
```

3 simplified application logic

```
in-context: off-hook do:
[ advertise: call (phone-call) on: phone (mobile-phone)
  [ play: phone call-waiting-signal during: 3 seconds ] ]
```

Basic Telephony Behaviour

2 straightforward application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ].
```

3 simplified application logic

```
in-context: off-hook do:
[ advertise: call (phone-call) on: phone (mobile-phone)
  [ play: phone call-waiting-signal during: 3 seconds ] ]
```

intrinsic to telephony

Basic Telephony Behaviour

application logic

```
receive: call (phone-call) on: phone (mobile-phone)
[ advertise: call on: phone.
  enqueue: call in: phone incoming-calls ].

advertise: call (phone-call) on: phone (mobile-phone)
[ play: phone ringtone during: 20 seconds ].
```

application logic

```
in-context:  off-hook  do:
[ advertise: call (phone-call) on: phone (mobile-phone)
  [ play: phone call-waiting-signal during: 3 seconds ] ]
```

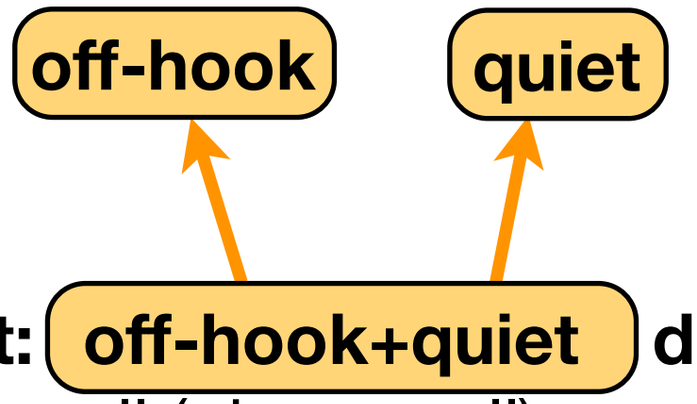
Context Combinations

in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

Context Combinations

in-context: {off-hook quiet} do:
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

Context Combinations

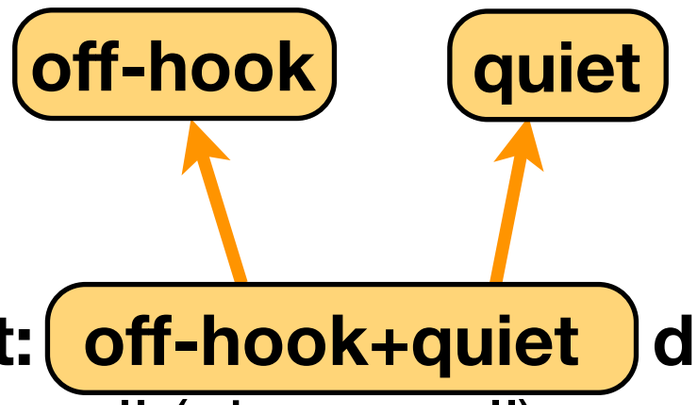


in-context: **off-hook+quiet** **do:**

[advertise: call (phone-call) on: phone (mobile-phone)

[...] **]**

Context Combinations



in-context: **off-hook+quiet** **do:**
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

The resulting combination is:

- Independent of order
- Unique

Context Combinations

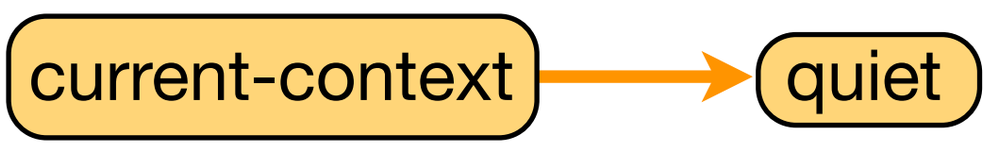
in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

current-context

Context Combinations

in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

activate-context: quiet

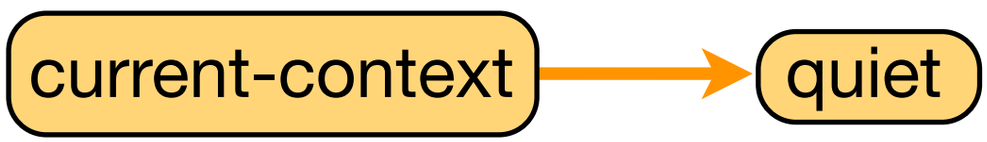


Context Combinations

in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

activate-context: quiet

activate-context: off-hook



Context Combinations

in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

activate-context: quiet

activate-context: off-hook

Context Combinations

1 dynamic behaviour adaptation

in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

activate-context: quiet

activate-context: off-hook

Context Combinations

in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[...] **]**

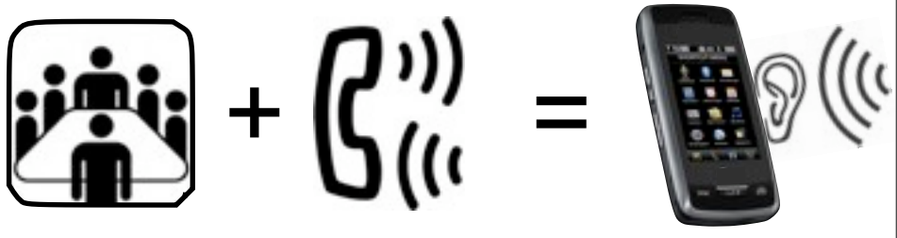
activate-context: quiet

activate-context: off-hook

Bypassing Resends

(avoiding context representation mismatches)

in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[**without-context:** quiet **do:**
[resend **]**] **]**

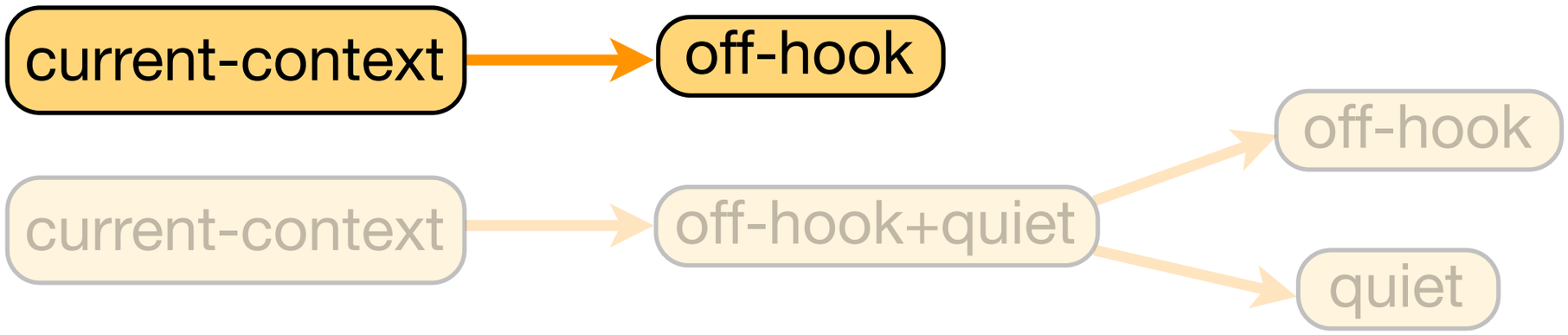


reminder

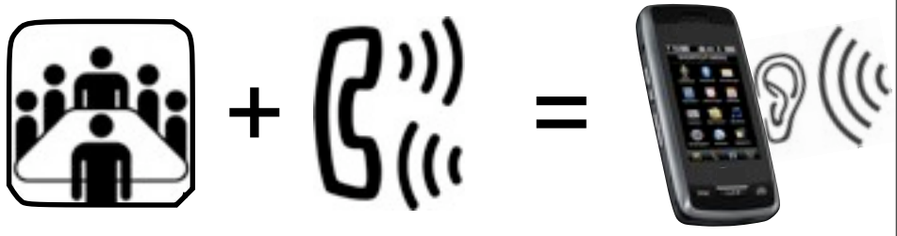
in-context: off-hook do:
[advertise: call (phone-call) on: phone (mobile-phone)
[play: phone call-waiting-signal during: 3 seconds] **]**

Bypassing Resends

(avoiding context representation mismatches)



in-context: { off-hook, ~~quiet~~ } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[**without-context:** quiet **do:**
[resend **]**] **]**

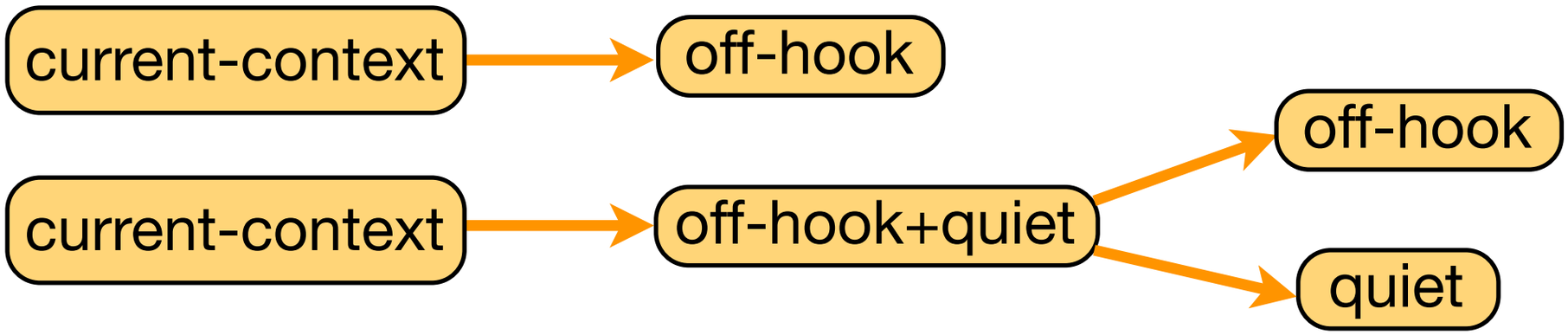


reminder

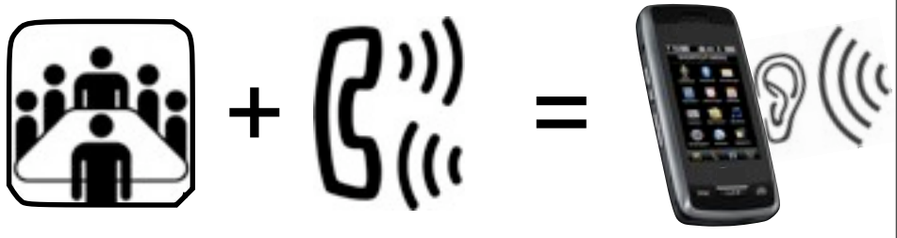
in-context: off-hook do:
[advertise: call (phone-call) on: phone (mobile-phone)
[play: phone call-waiting-signal during: 3 seconds] **]**

Bypassing Resends

(avoiding context representation mismatches)



in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[**without-context:** quiet **do:**
[resend **]**] **]**



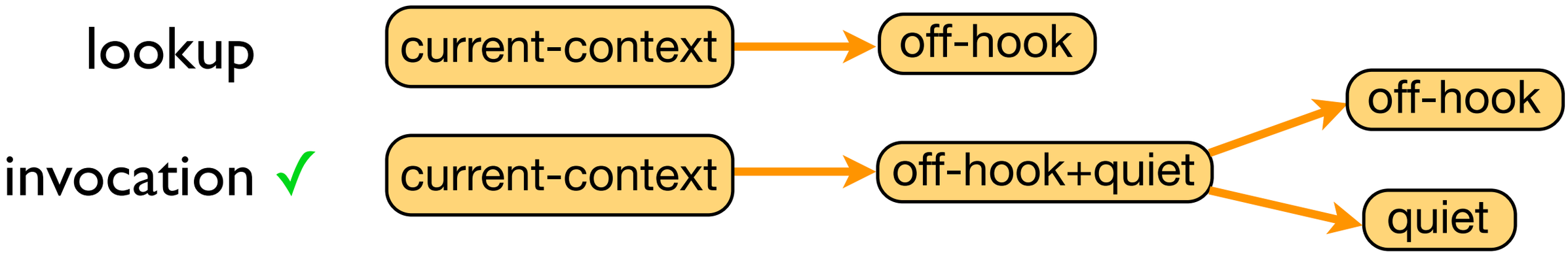
in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[resend-bypassing: quiet] **]**

reminder

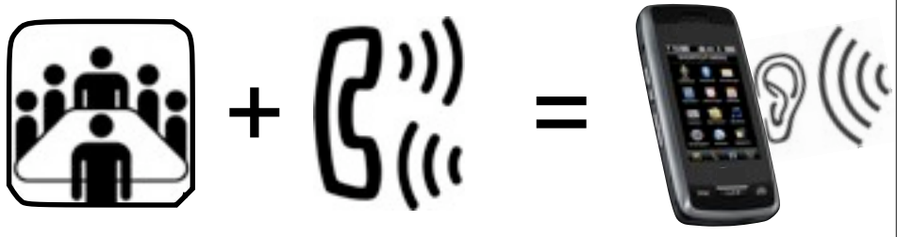
in-context: off-hook do:
[advertise: call (phone-call) on: phone (mobile-phone)
[play: phone call-waiting-signal during: 3 seconds] **]**

Bypassing Resends

(avoiding context representation mismatches)



in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[**without-context:** quiet **do:**
[resend **]**] **]**



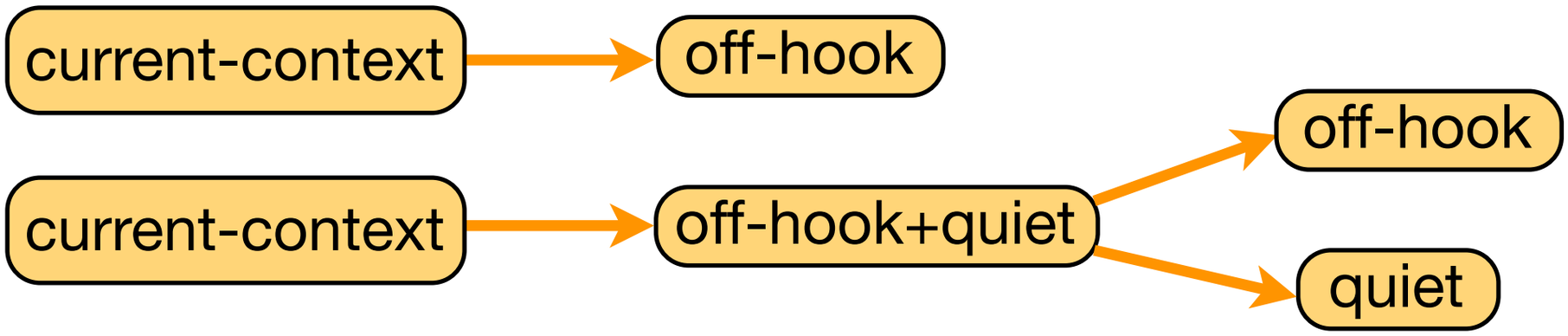
in-context: { off-hook, ~~quiet~~ } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[resend-bypassing: quiet] **]**

reminder

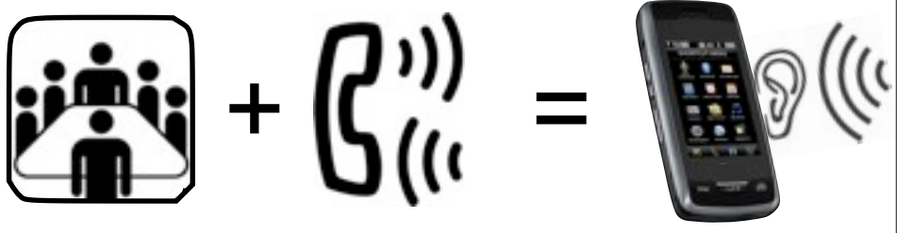
in-context: off-hook do:
[advertise: call (phone-call) on: phone (mobile-phone)
[play: phone call-waiting-signal during: 3 seconds] **]**

Bypassing Resends

1 dynamic behaviour adaptation



in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[**without-context:** quiet **do:**
[resend **]**] **]**



in-context: { off-hook, quiet } do:
[advertise: call (phone-call) on: phone (mobile-phone)
[resend-bypassing: quiet] **]**

reminder

in-context: off-hook do:
[advertise: call (phone-call) on: phone (mobile-phone)
[play: phone call-waiting-signal during: 3 seconds] **]**

Coherent Context Activation

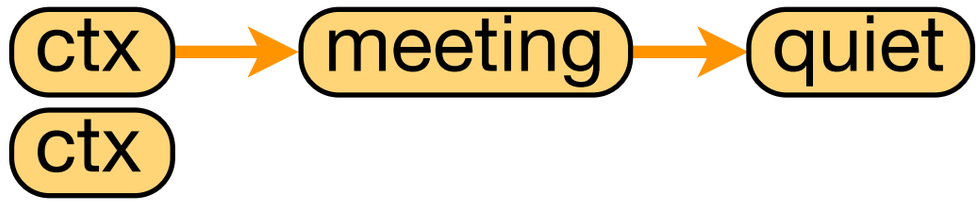
Coherent Context Activation

activate-context: meeting ✓ quiet

Coherent Context Activation

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet

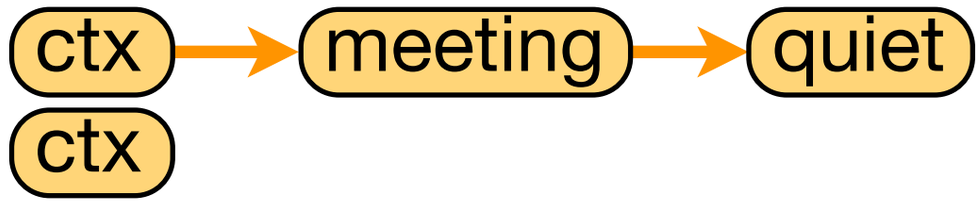


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet

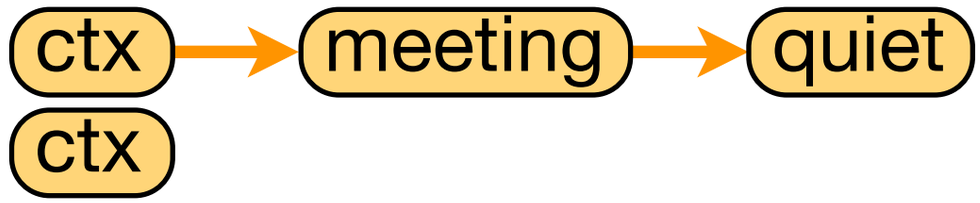


Coherent Context Activation

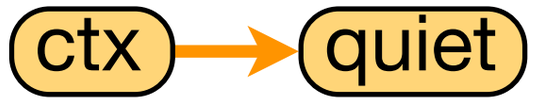
induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



activate-context: quiet ✓ quiet

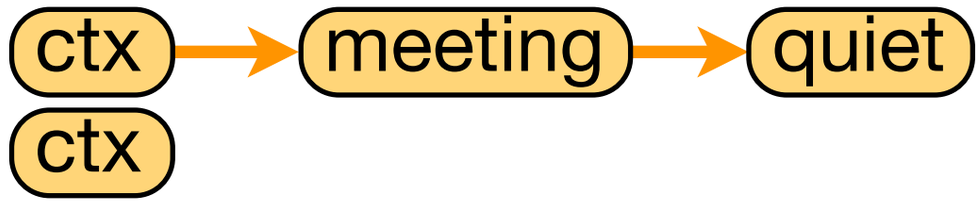


Coherent Context Activation

induced

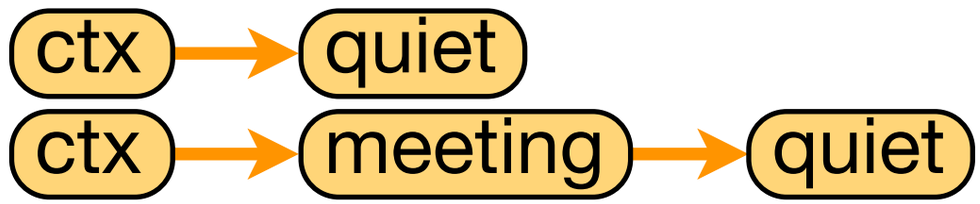
activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



activate-context: quiet ✓ quiet

activate-context: meeting

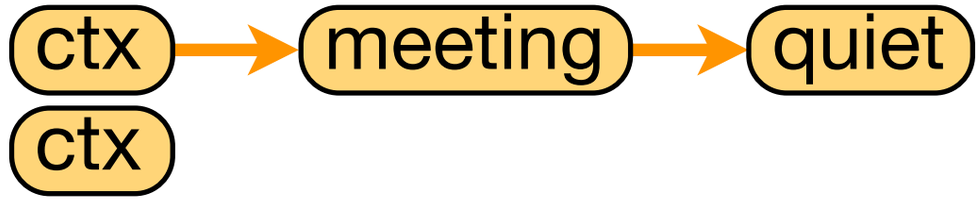


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

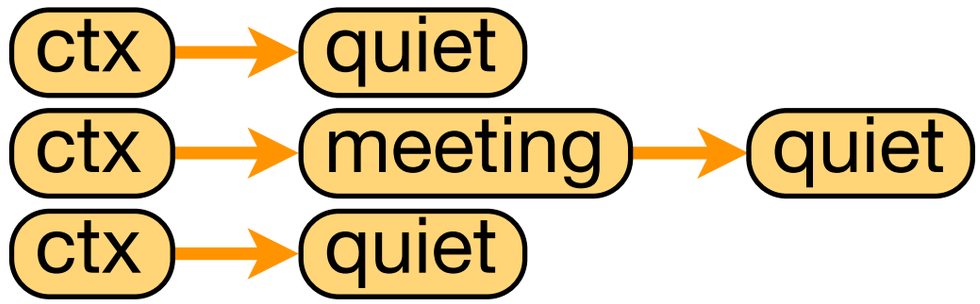
deactivate-context: meeting ✗ quiet



activate-context: quiet ✓ quiet

activate-context: meeting

deactivate-context: meeting

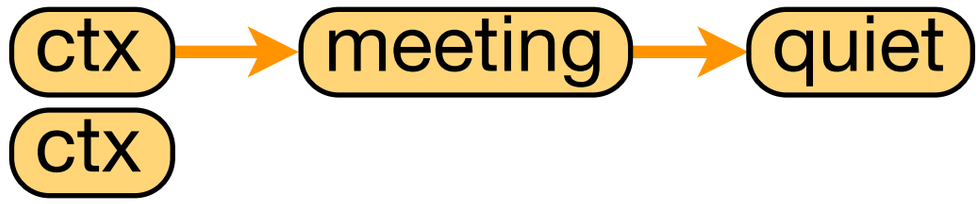


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet

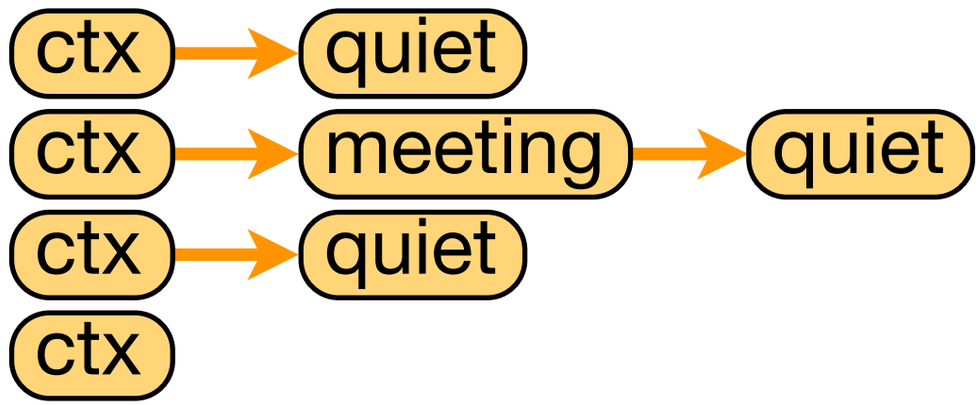


activate-context: quiet ✓ quiet

activate-context: meeting

deactivate-context: meeting

deactivate-context: quiet ✗ quiet

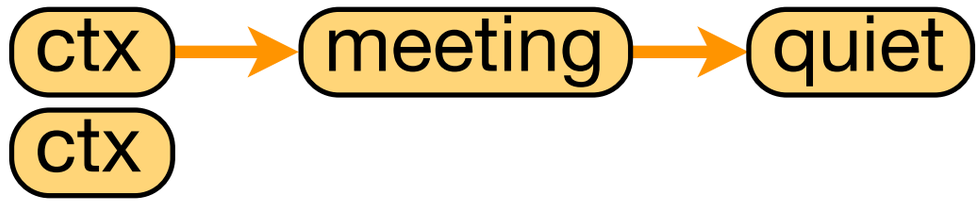


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



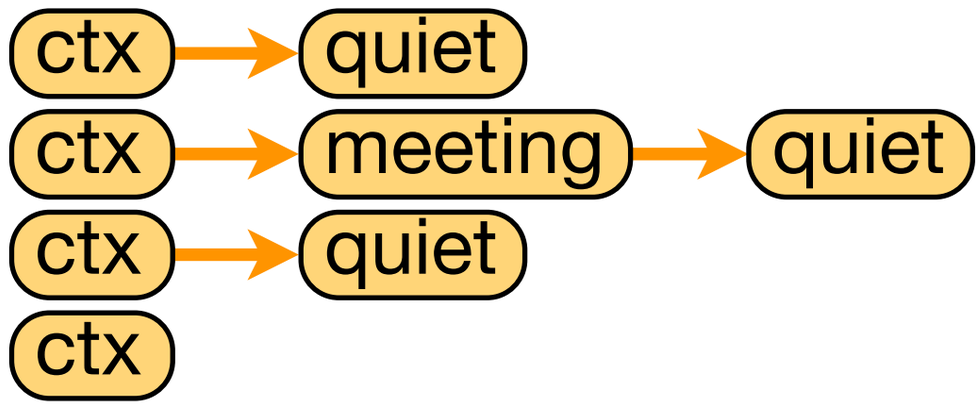
nested

activate-context: quiet ✓ quiet

activate-context: meeting

deactivate-context: meeting

deactivate-context: quiet ✗ quiet

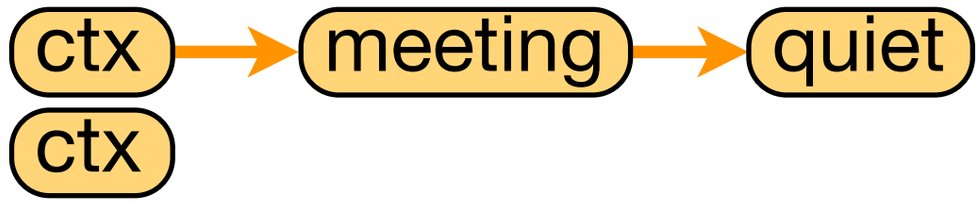


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



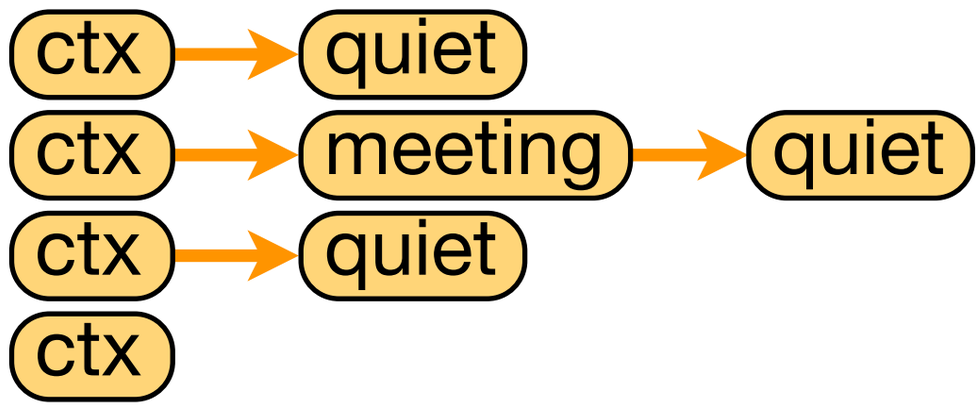
nested

activate-context: quiet ✓ quiet

activate-context: meeting

deactivate-context: meeting

deactivate-context: quiet ✗ quiet

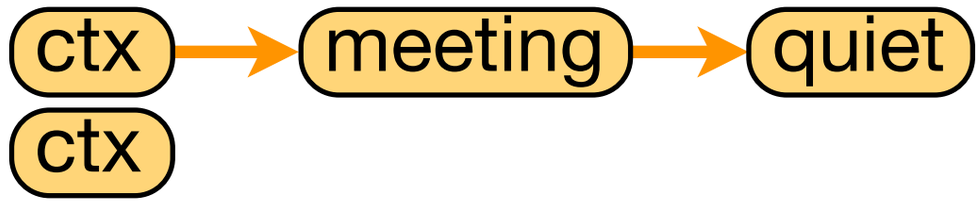


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



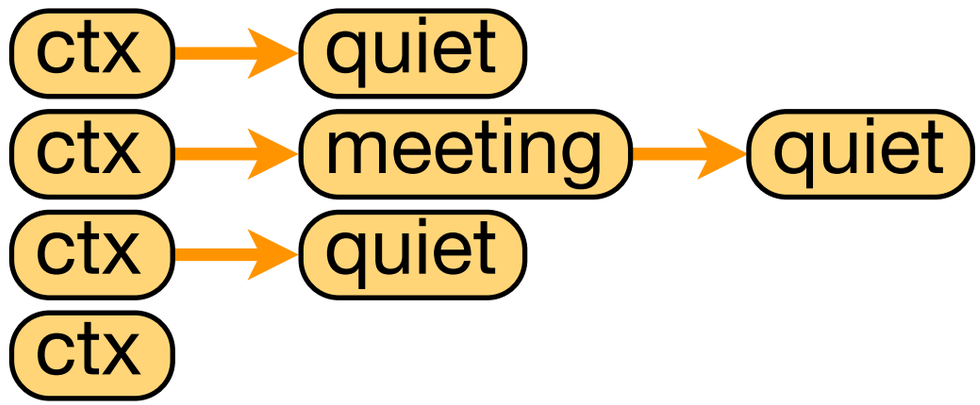
nested

activate-context: quiet ✓ quiet

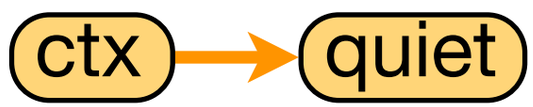
activate-context: meeting

deactivate-context: meeting

deactivate-context: quiet ✗ quiet



activate-context: quiet ✓ quiet

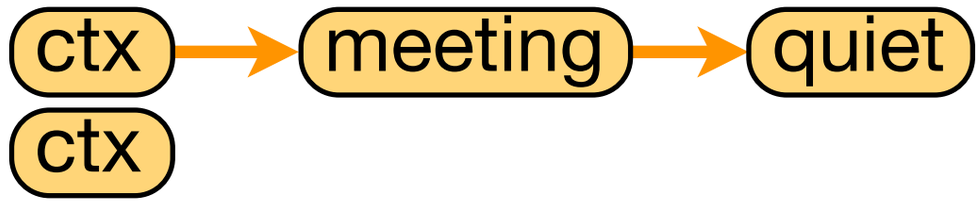


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



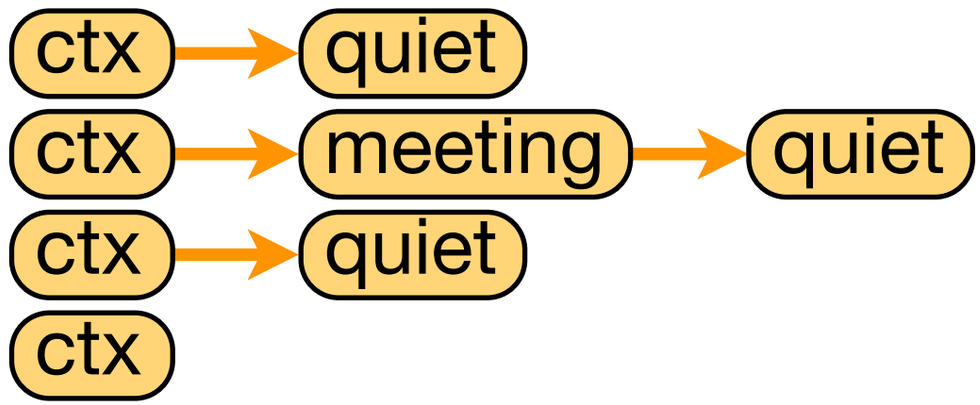
nested

activate-context: quiet ✓ quiet

activate-context: meeting

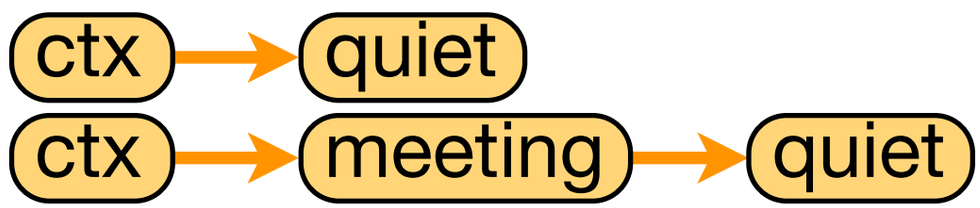
deactivate-context: meeting

deactivate-context: quiet ✗ quiet



activate-context: quiet ✓ quiet

activate-context: meeting

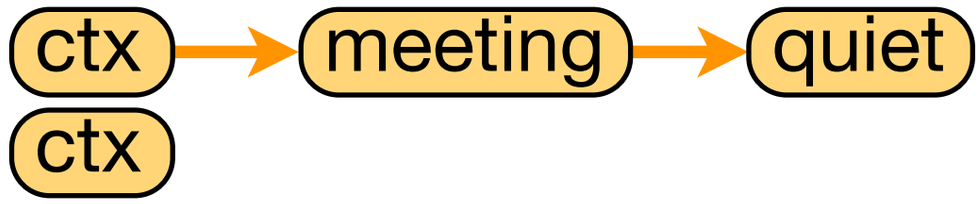


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



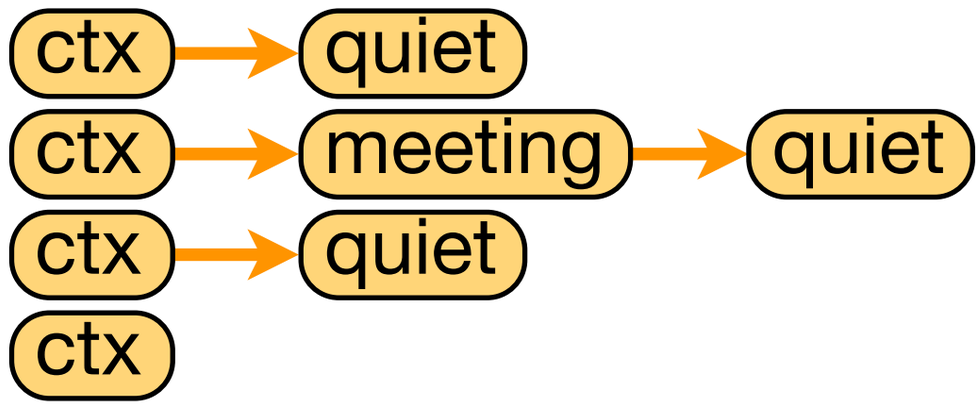
nested

activate-context: quiet ✓ quiet

activate-context: meeting

deactivate-context: meeting

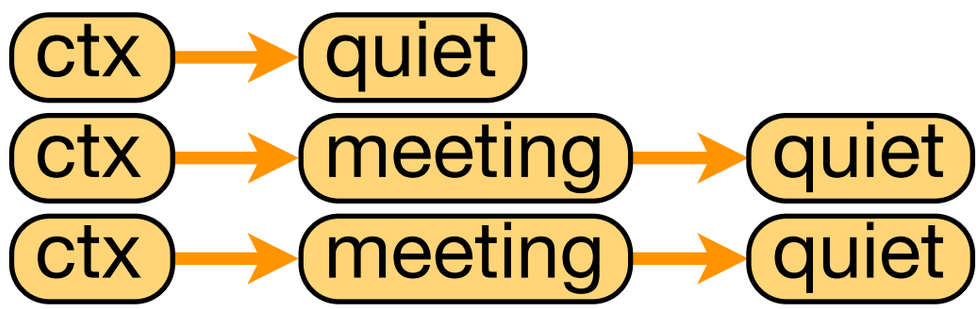
deactivate-context: quiet ✗ quiet



activate-context: quiet ✓ quiet

activate-context: meeting

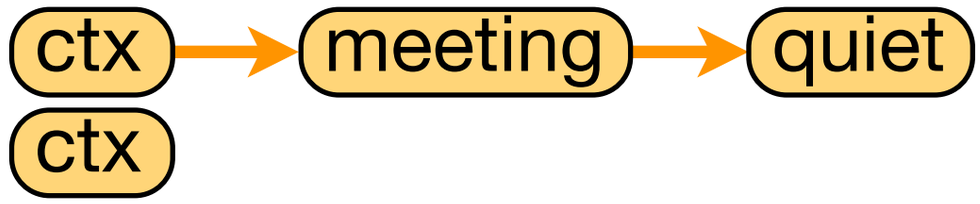
deactivate-context: quiet



Coherent Context Activation

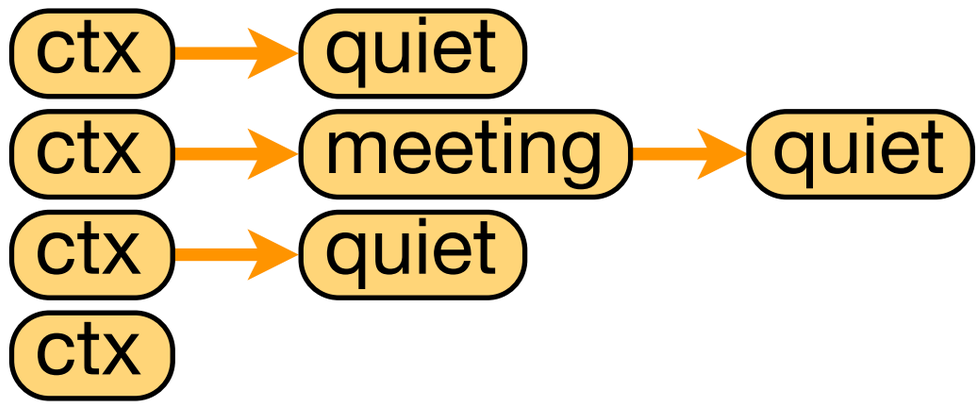
induced

activate-context: meeting	✓ quiet
deactivate-context: meeting	✗ quiet

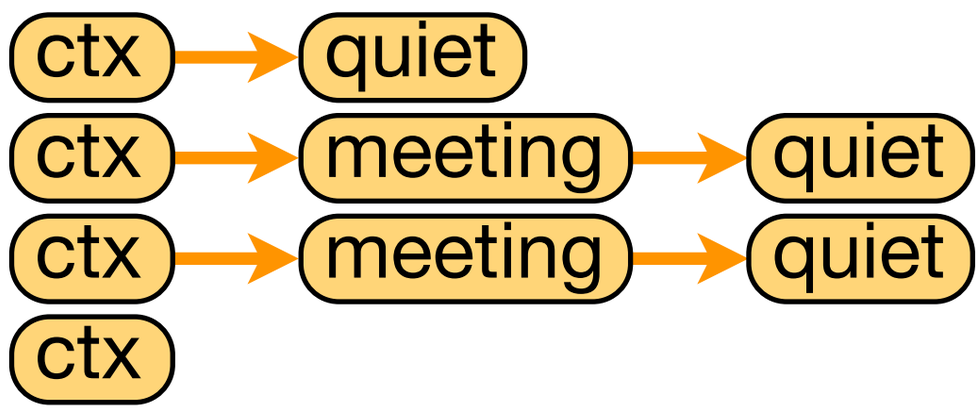


nested

activate-context: quiet	✓ quiet
activate-context: meeting	
deactivate-context: quiet	
deactivate-context: meeting	✗ quiet



activate-context: quiet	✓ quiet
activate-context: meeting	
deactivate-context: quiet	
deactivate-context: meeting	✗ quiet

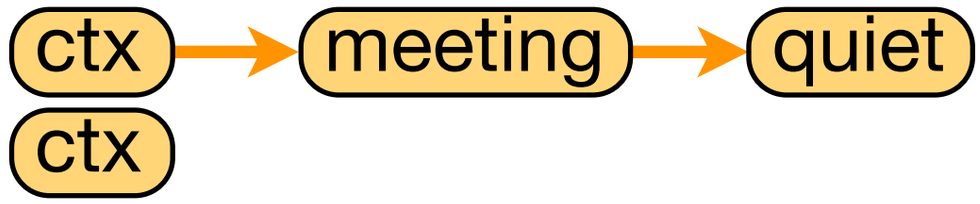


Coherent Context Activation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



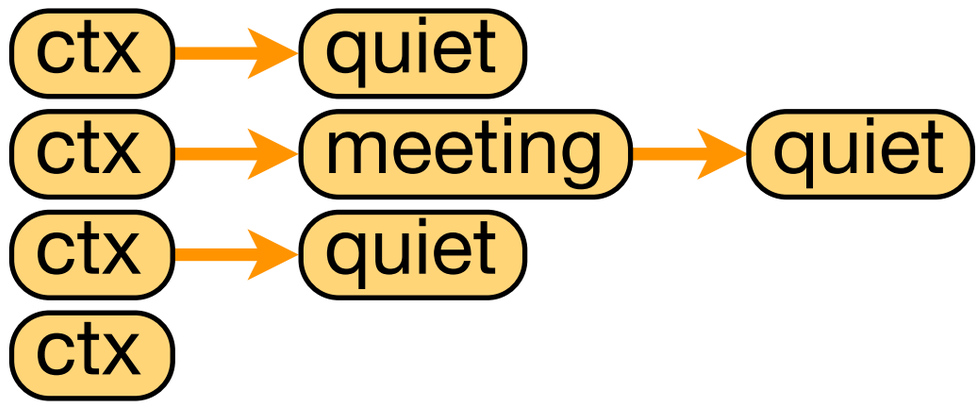
nested

activate-context: quiet ✓ quiet

activate-context: meeting

deactivate-context: meeting

deactivate-context: quiet ✗ quiet



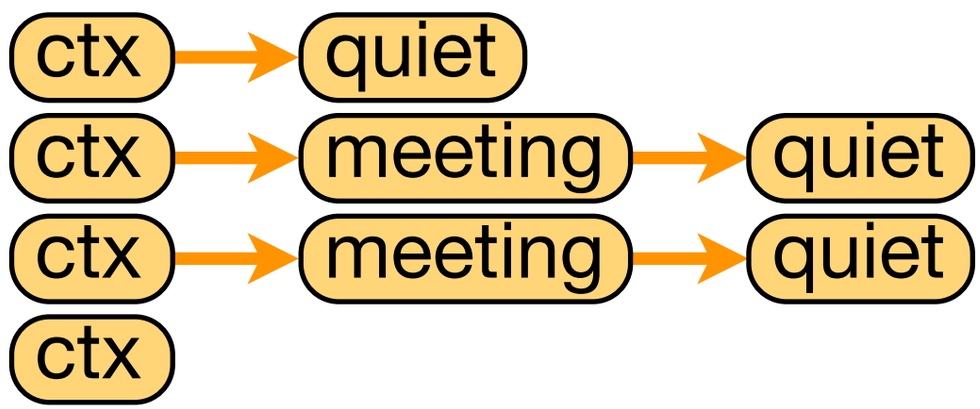
interleaved

activate-context: quiet ✓ quiet

activate-context: meeting

deactivate-context: quiet

deactivate-context: meeting ✗ quiet

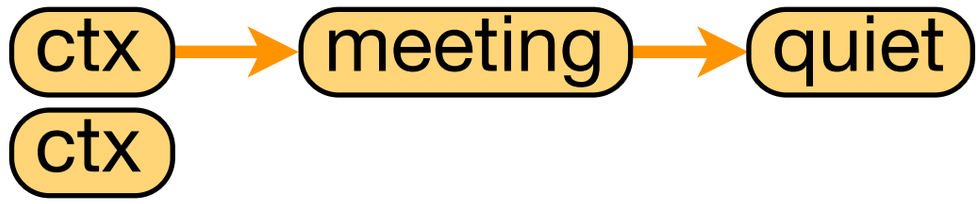


Coherent Context Activation

induced

activate-context: meeting
deactivate-context: meeting

✓ quiet
✗ quiet



nested

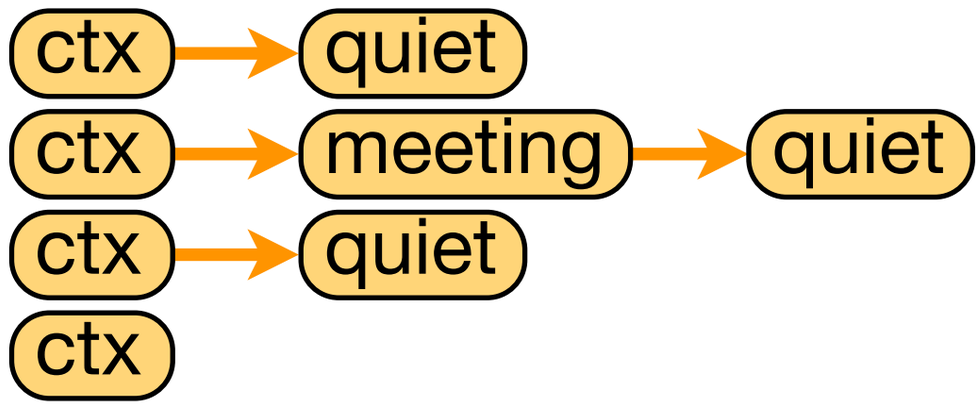
activate-context: quiet

✓ quiet

activate-context: meeting
deactivate-context: meeting

deactivate-context: quiet

✗ quiet



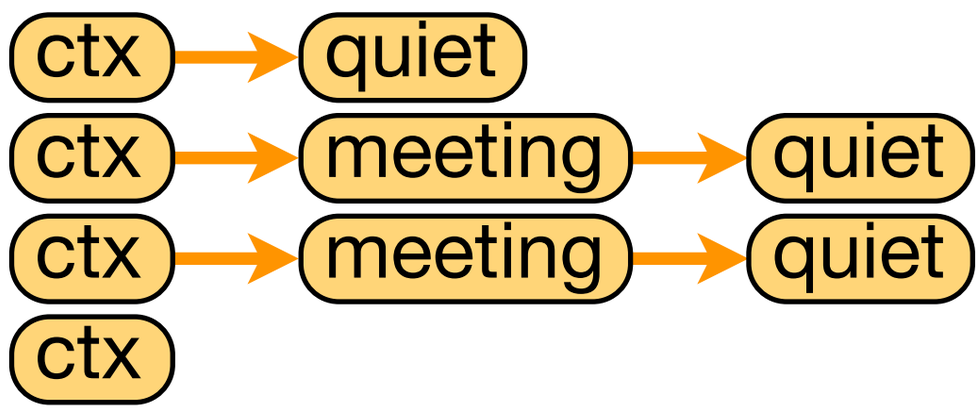
interleaved

activate-context: quiet

✓ quiet

activate-context: meeting
deactivate-context: quiet
deactivate-context: meeting

✗ quiet



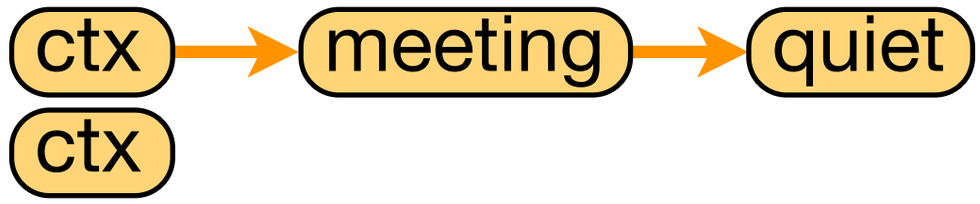
Coherent Context Activation

1 dynamic behaviour adaptation

induced

activate-context: meeting ✓ quiet

deactivate-context: meeting ✗ quiet



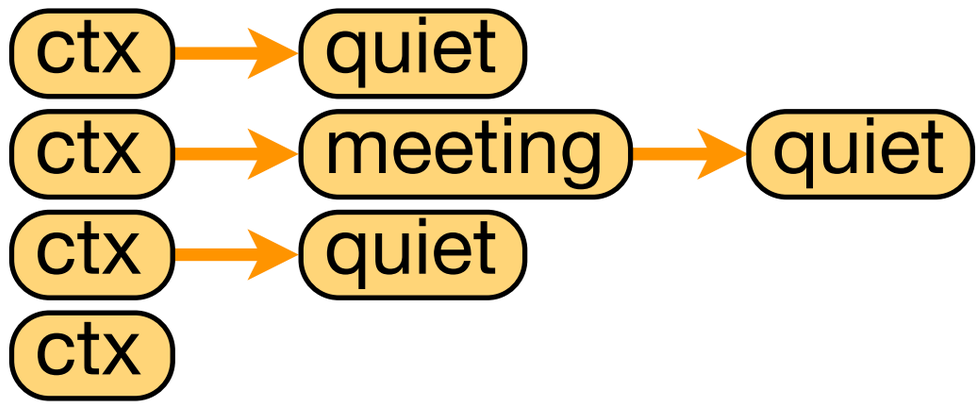
nested

activate-context: quiet ✓ quiet

activate-context: meeting

deactivate-context: meeting

deactivate-context: quiet ✗ quiet



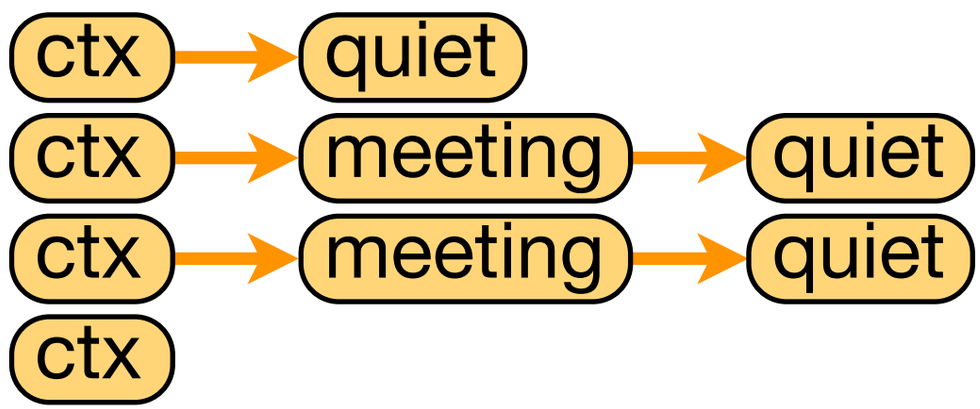
interleaved

activate-context: quiet ✓ quiet

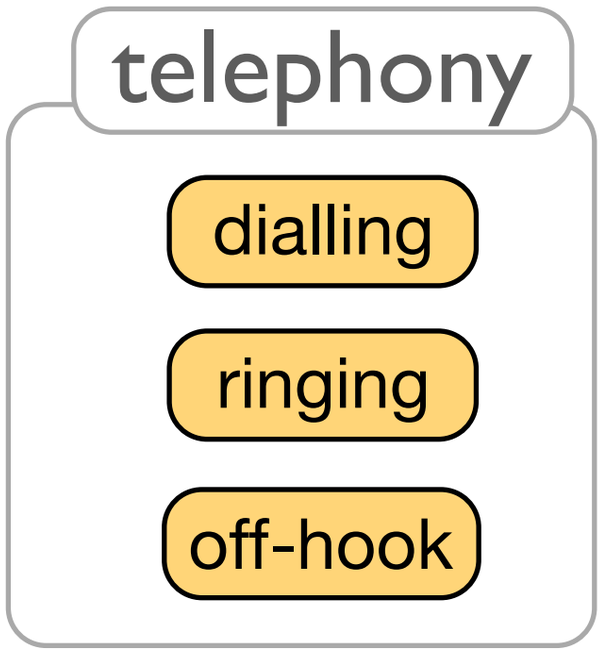
activate-context: meeting

deactivate-context: quiet

deactivate-context: meeting ✗ quiet



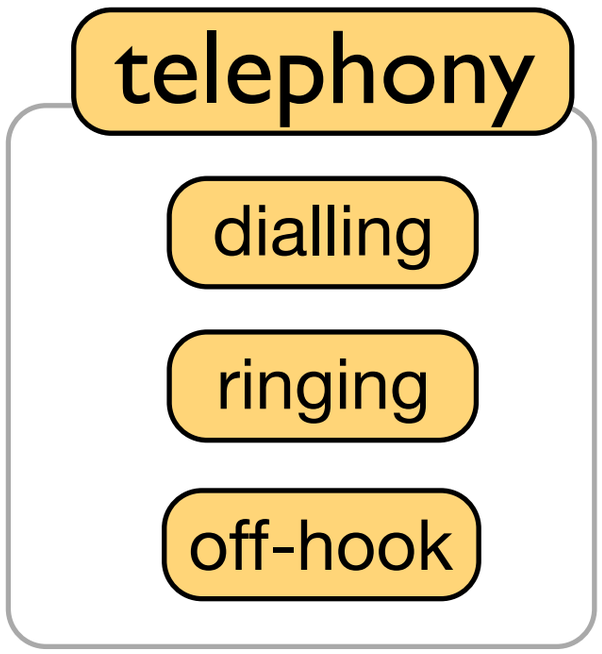
Contexts as Software Features



advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds]

in-context: off-hook do:
[advertise: call (phone-call) on: phone (mobile-phone)
[play: phone call-waiting-signal during: 3 seconds] **]**

Contexts as Software Features



advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds]

in-context: off-hook do:
[advertise: call (phone-call) on: phone (mobile-phone)
[play: phone call-waiting-signal during: 3 seconds] **]**

Contexts as Software Features

```
in-context: telephony do:
[
        define: dialling as: context clone.

        define: ringing as: context clone.

        define: off-hook as: context clone.


        advertise: call (phone-call) on: phone (mobile-phone)
        [ play: phone ringtone during: 20 seconds ]

        in-context: off-hook do:
        [ advertise: call (phone-call) on: phone (mobile-phone)
          [ play: phone call-waiting-signal during: 3 seconds ] ]
]
```

Contexts as Software Features

framework context

telephony

define: dialling as: context clone.

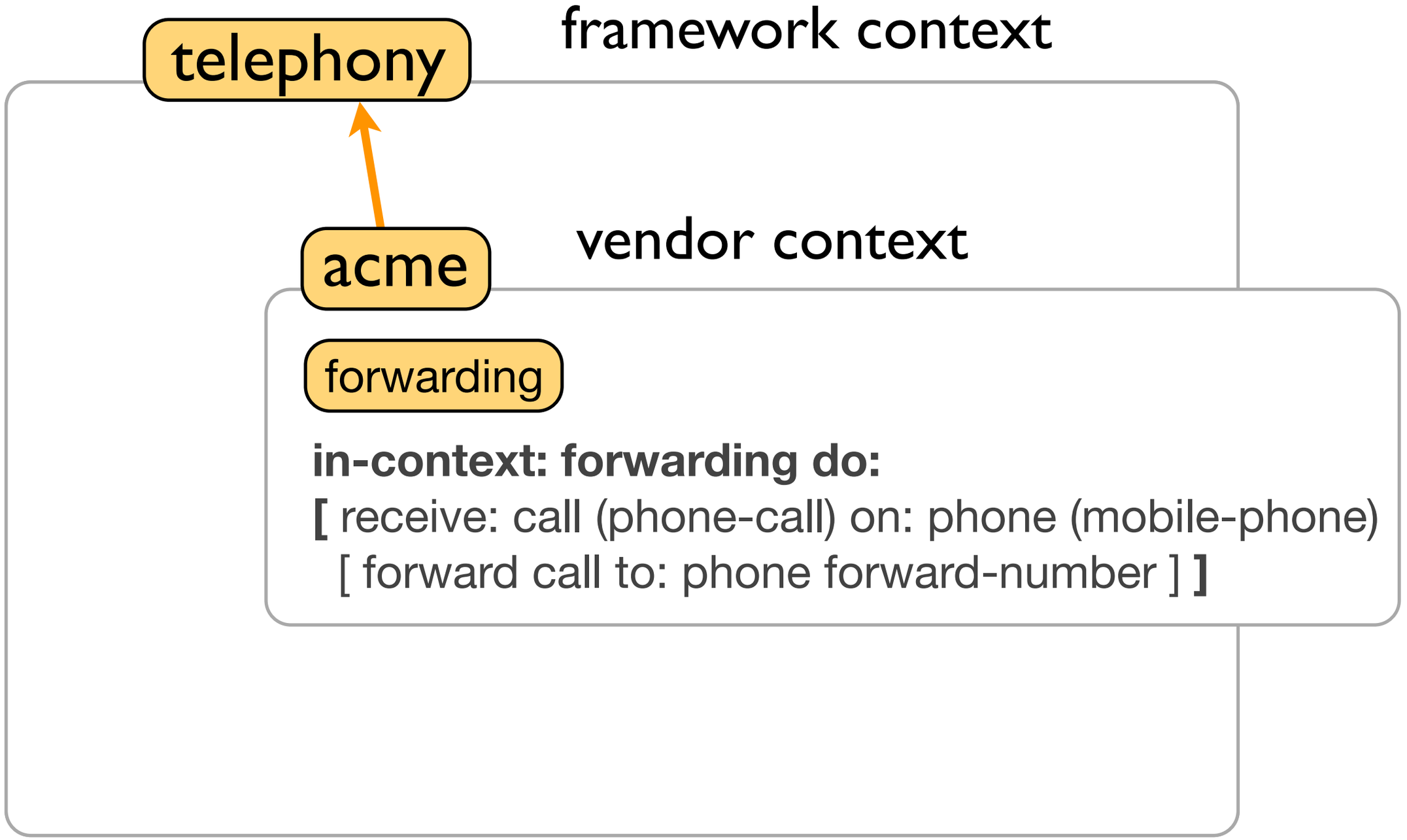
define: ringing as: context clone.

define: off-hook as: context clone.

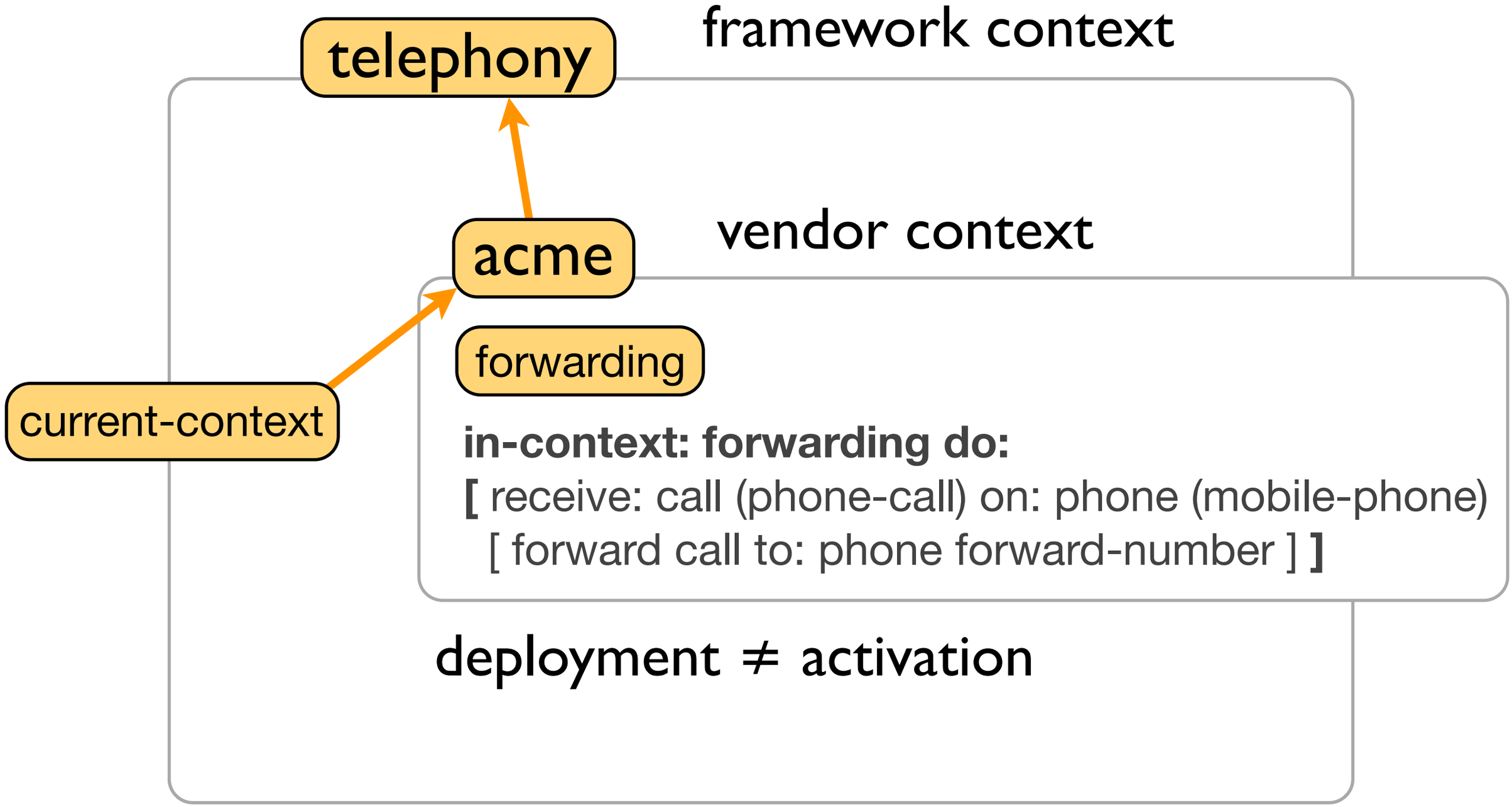
advertise: call (phone-call) on: phone (mobile-phone)
[play: phone ringtone during: 20 seconds]

in-context: off-hook do:
[advertise: call (phone-call) on: phone (mobile-phone)
[play: phone call-waiting-signal during: 3 seconds] **]**

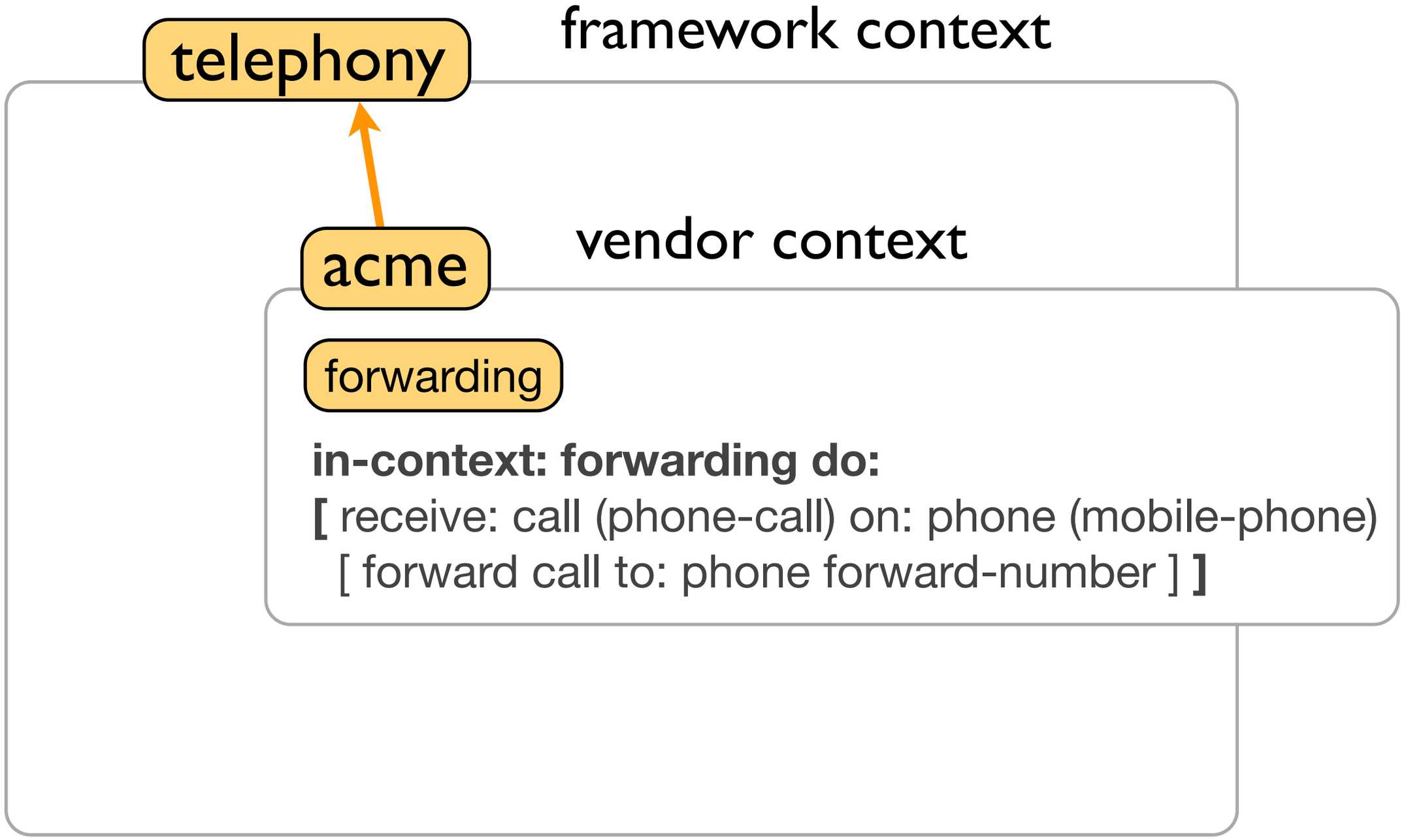
Contexts as Software Features



Contexts as Software Features

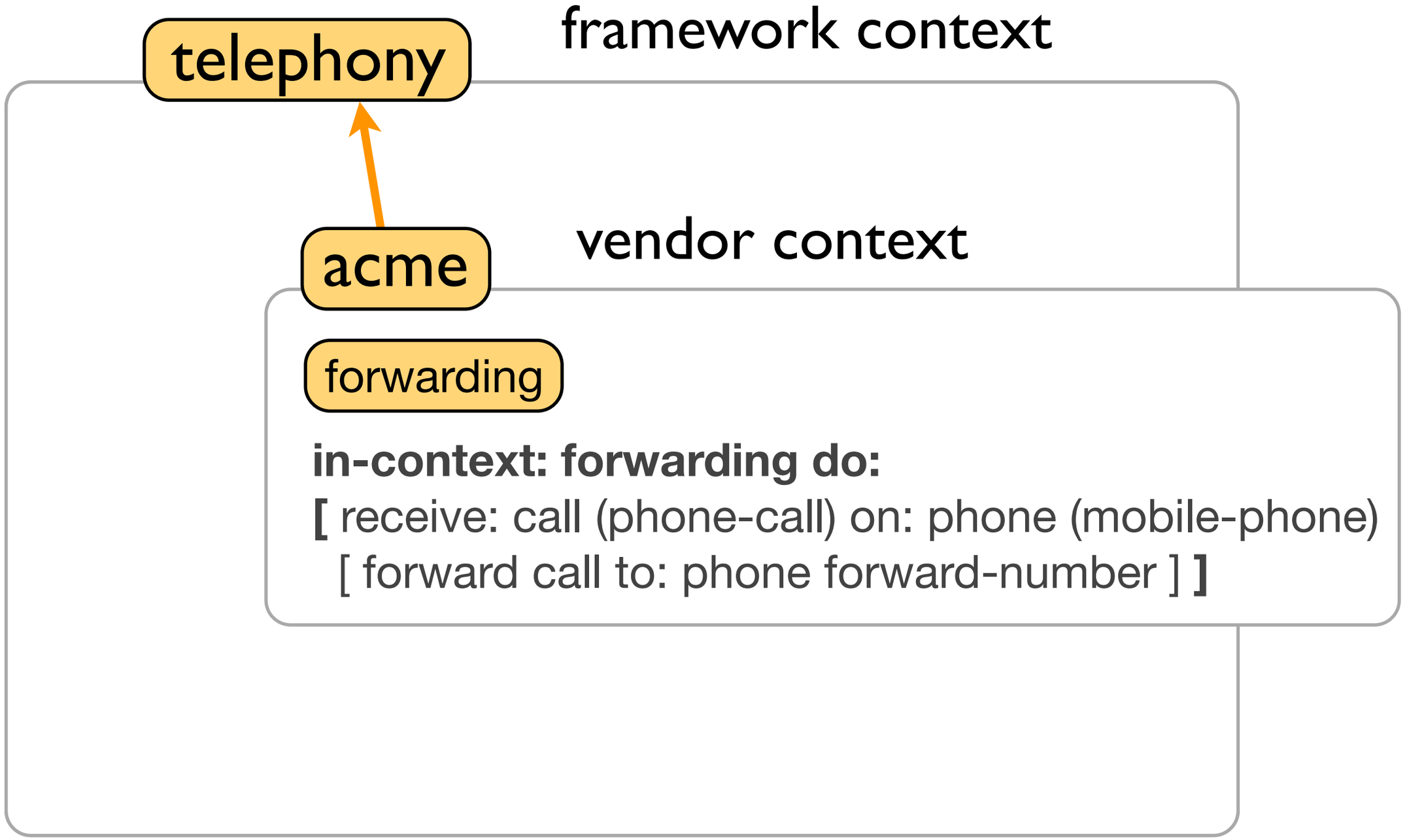


Contexts as Software Features



5 straightforward software architectures

Contexts as Software Features



Implementation

Ambience

C, Flex, Bison, Common Lisp

AmOS

Common Lisp

- Open implementation
- Meta Object Protocol
- Context Object Protocol

Semantics

PLT Redex

- Operational, small step
- Executable
- Graphical browsers

Conclusion

gearing up for dynamic context adaptation

Conclusion

gearing up for dynamic context adaptation

Context-Oriented Programming

Subjective Programming

Multimethods

Prototypes

Conclusion

gearing up for dynamic context adaptation

1. dynamic behaviour adaptation to context

- Context-Oriented Programming
- Subjective Programming
- Multimethods
- Prototypes

Conclusion

gearing up for dynamic context adaptation

1 2 straightforward application logic

Context-Oriented Programming

Subjective Programming

Multimethods

Prototypes

Conclusion

gearing up for dynamic context adaptation

1 2 3 simplified application logic

Context-Oriented Programming

Subjective Programming

Multimethods

Prototypes

Conclusion

gearing up for dynamic context adaptation

1 2 3 4 non-intrusive adaptations

- Context-Oriented Programming
- Subjective Programming
- Multimethods
- Prototypes

Conclusion

gearing up for dynamic context adaptation

straightforward architectures

Context-Oriented Programming

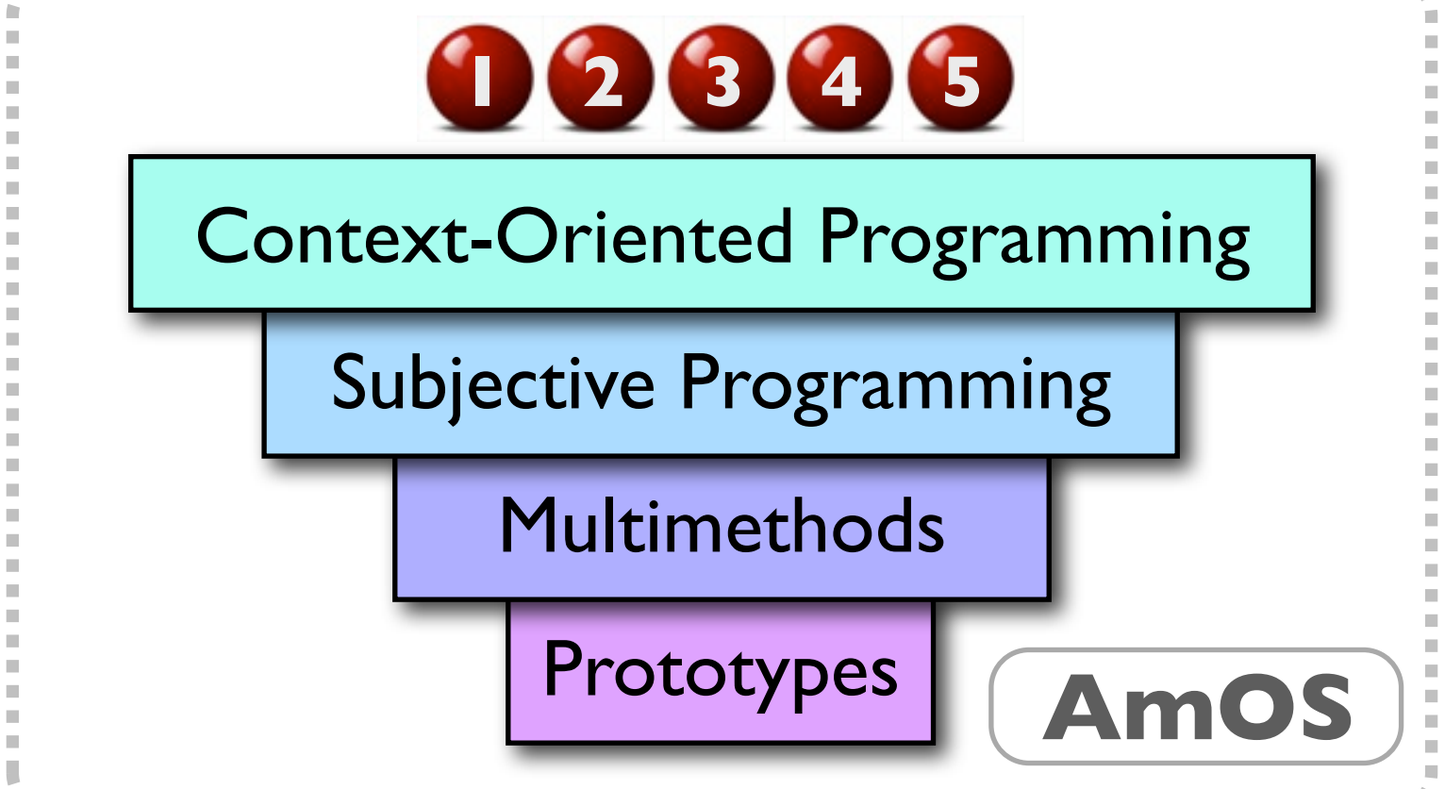
Subjective Programming

Multimethods

Prototypes

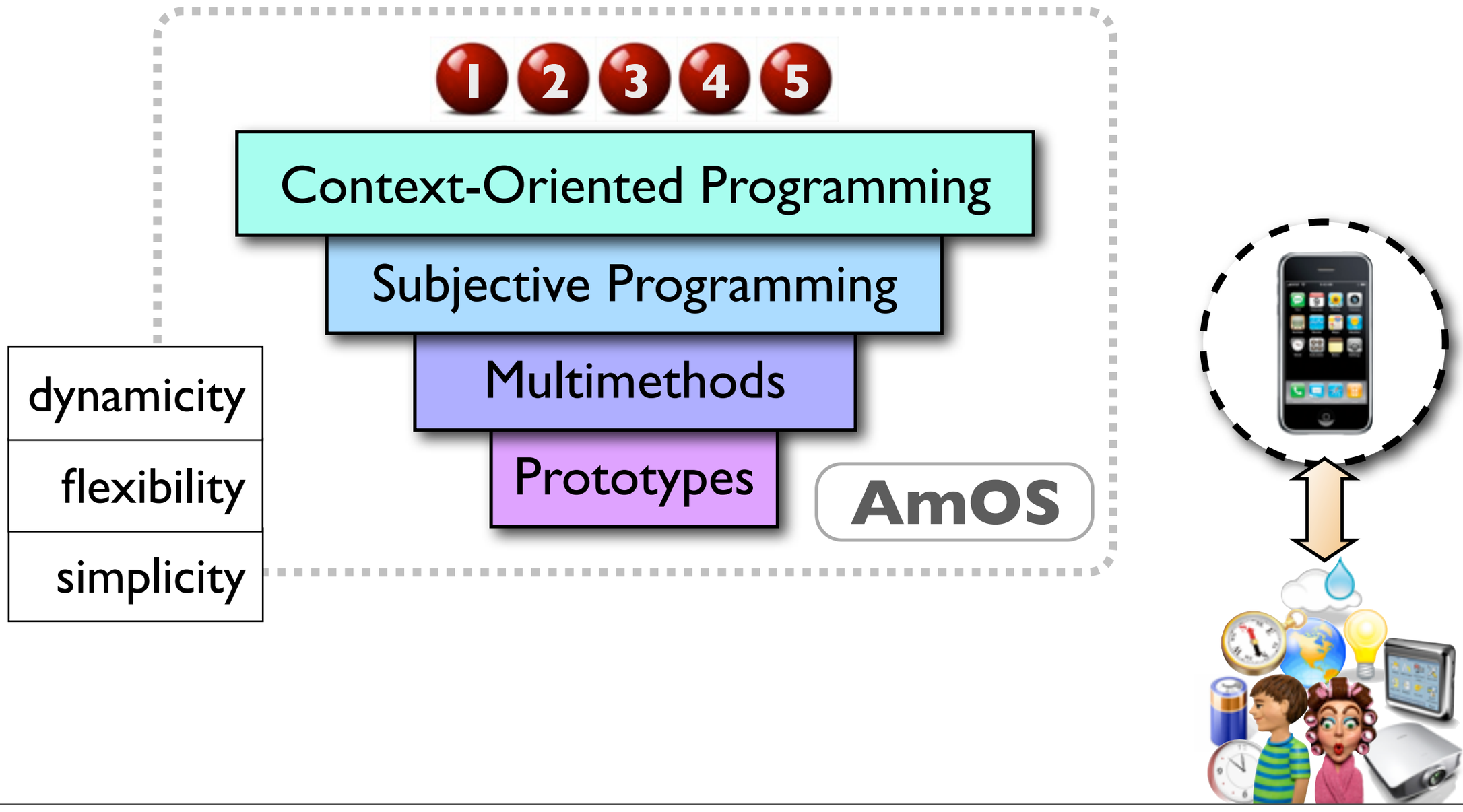
Conclusion

gearing up for dynamic context adaptation



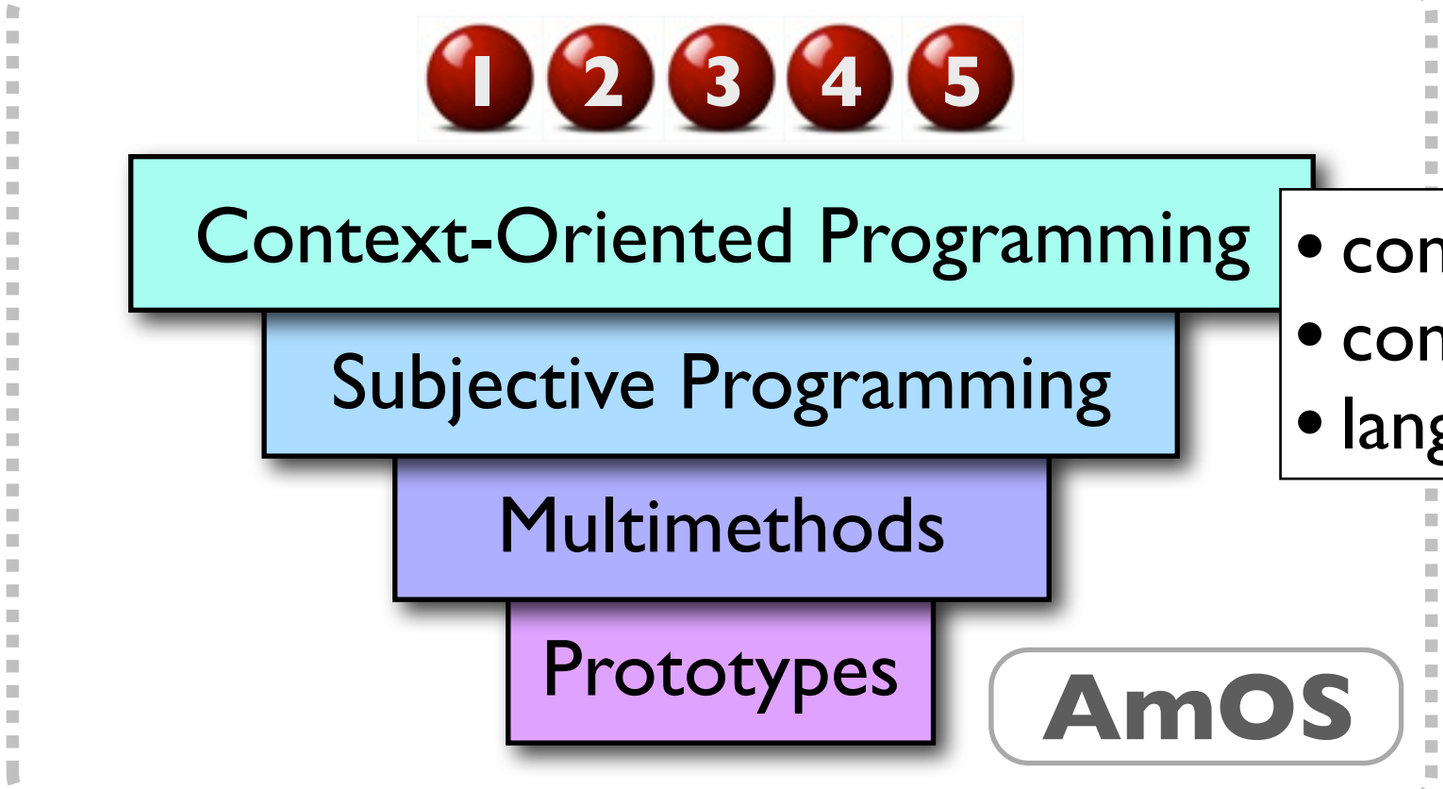
Conclusion

gearing up for dynamic context adaptation



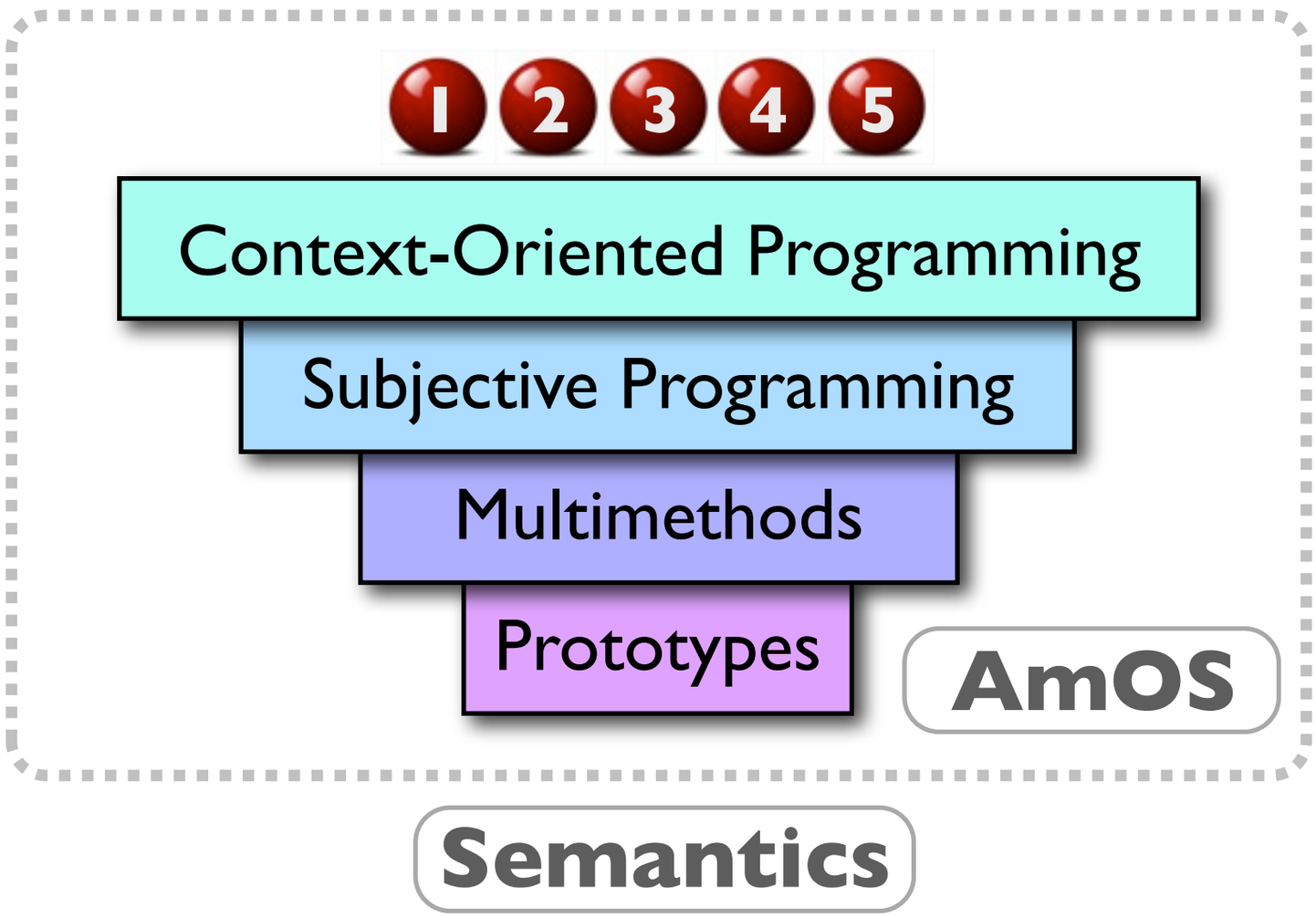
Conclusion

gearing up for dynamic context adaptation



Conclusion

gearing up for dynamic context adaptation



Conclusion

gearing up for dynamic context adaptation

matching syntax
tuned for dynamicity

Ambience

1 2 3 4 5

Context-Oriented Programming

Subjective Programming

Multimethods

Prototypes

AmOS

Semantics

Future Work

Context-Oriented Programming

- more extensive and realistic validation
- design and programming methodologies
- further context management techniques
- context lifetime
- development tools

Ambient-Oriented Programming

- concurrency support
- distribution support
- security

programming in

Ambience

gearing up for dynamic adaptation to context

Time for questions

Thank you for your attention

http://ambience.info.ucl.ac.be