Ambient References

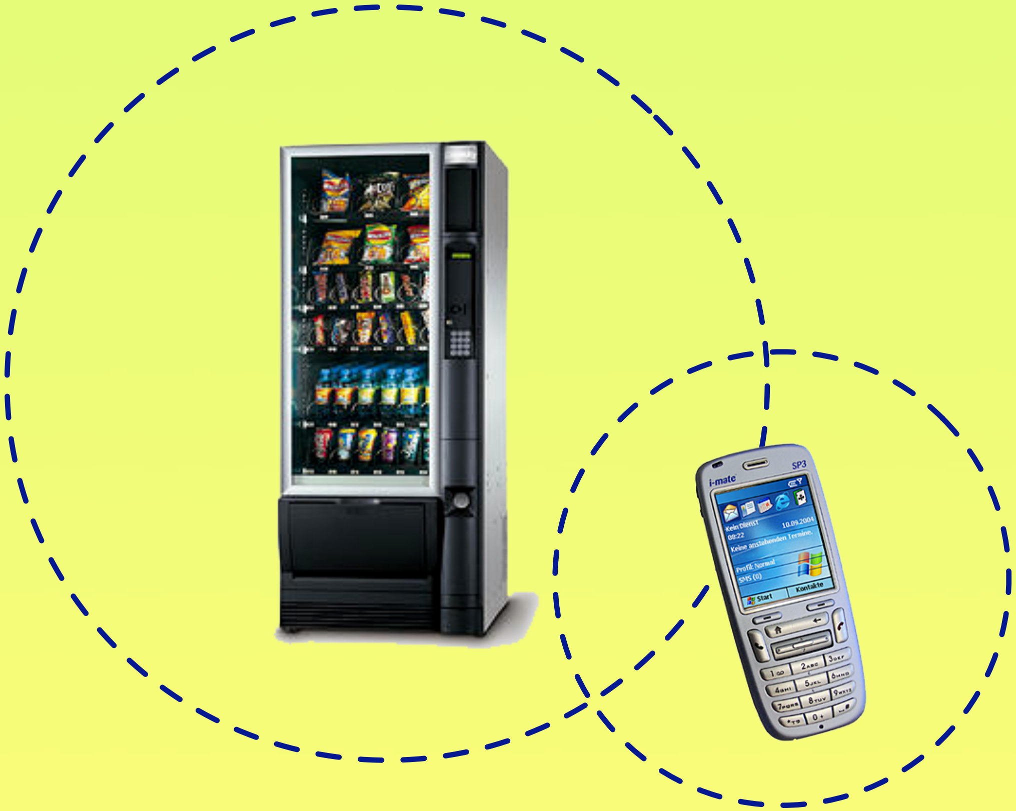
Addressing Objects in Mobile Networks

Tom Van Cutsem

Programming Technology Laboratory
Vrije Universiteit Brussel

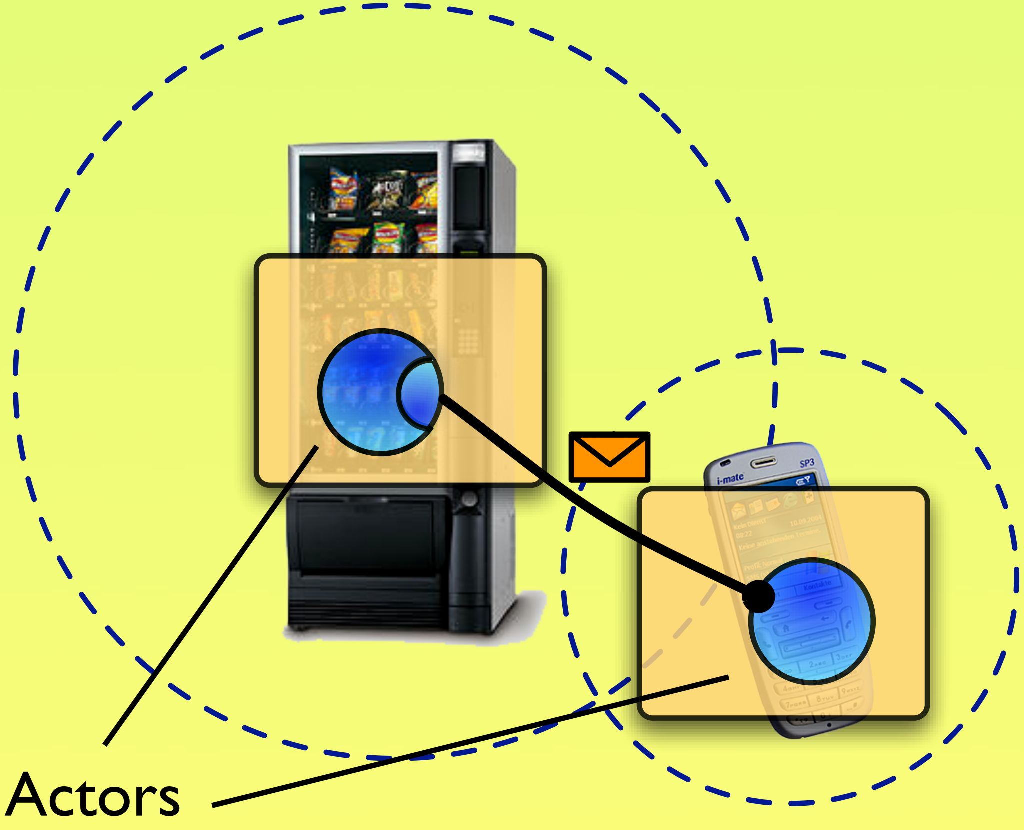
Mobile Networks

Pervasive/Ubiquitous Computing



Mobile Networks

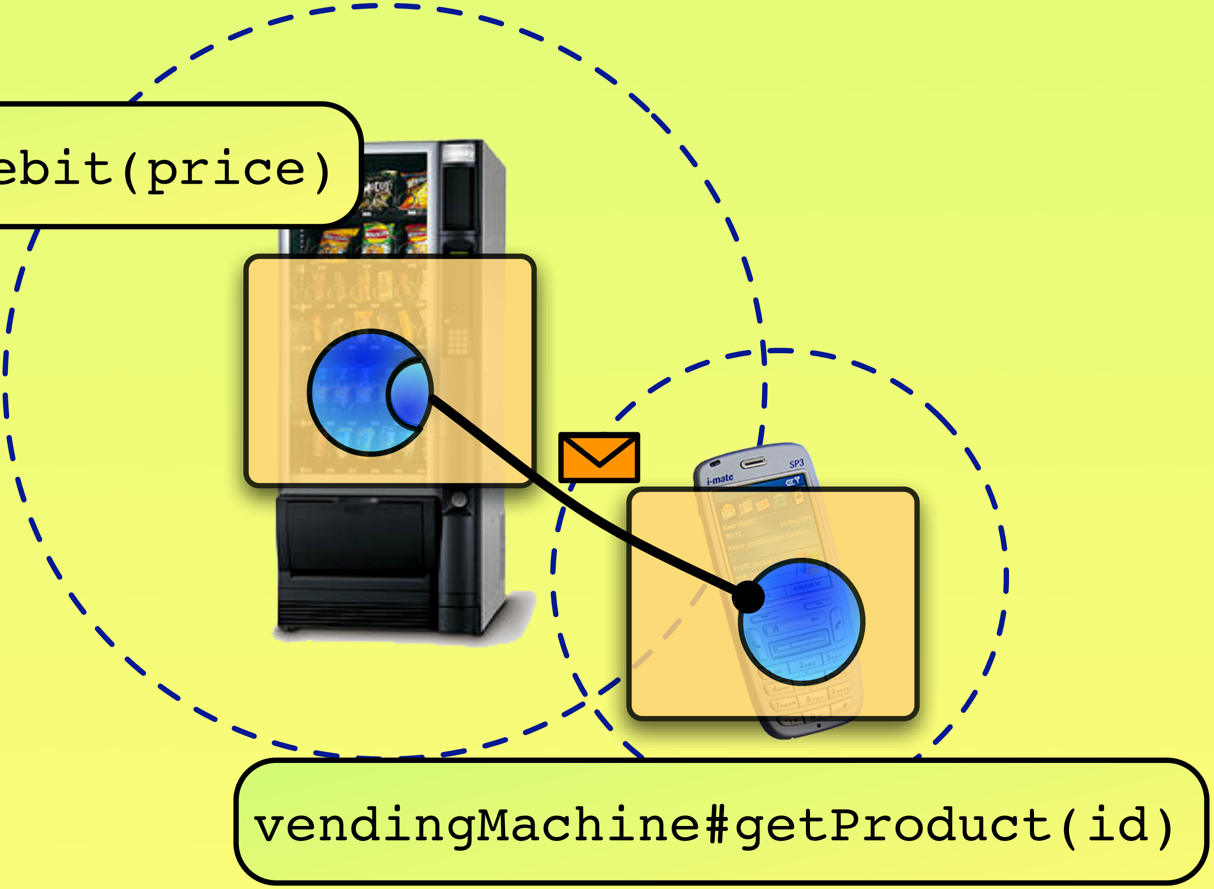
Pervasive/Ubiquitous Computing



Actors

Mobile Networks

Pervasive/Ubiquitous
Computing



```
cellPhone#debit(price)
```

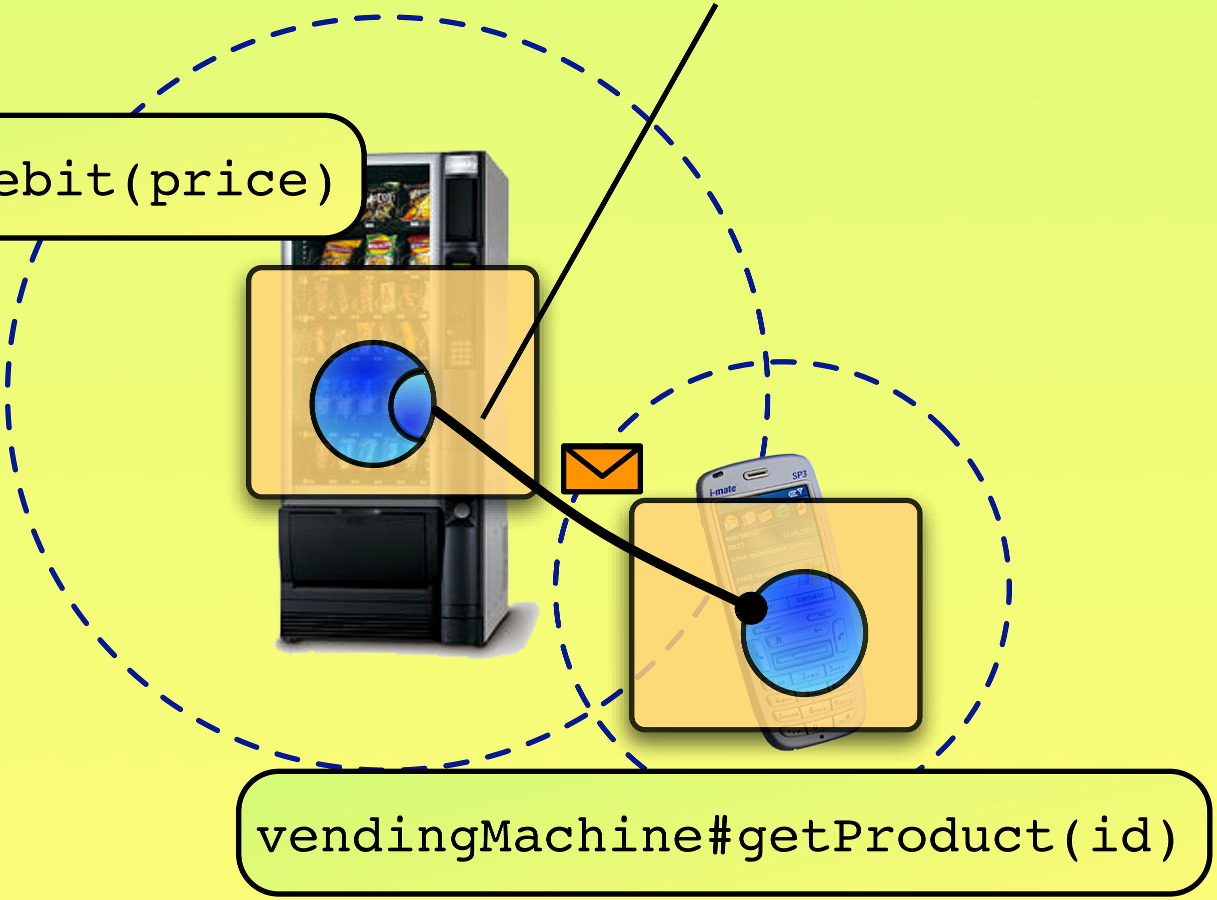
```
vendingMachine#getProduct(id)
```

Mobile Networks

Pervasive/Ubiquitous Computing

Ambient References

"remote actor references"



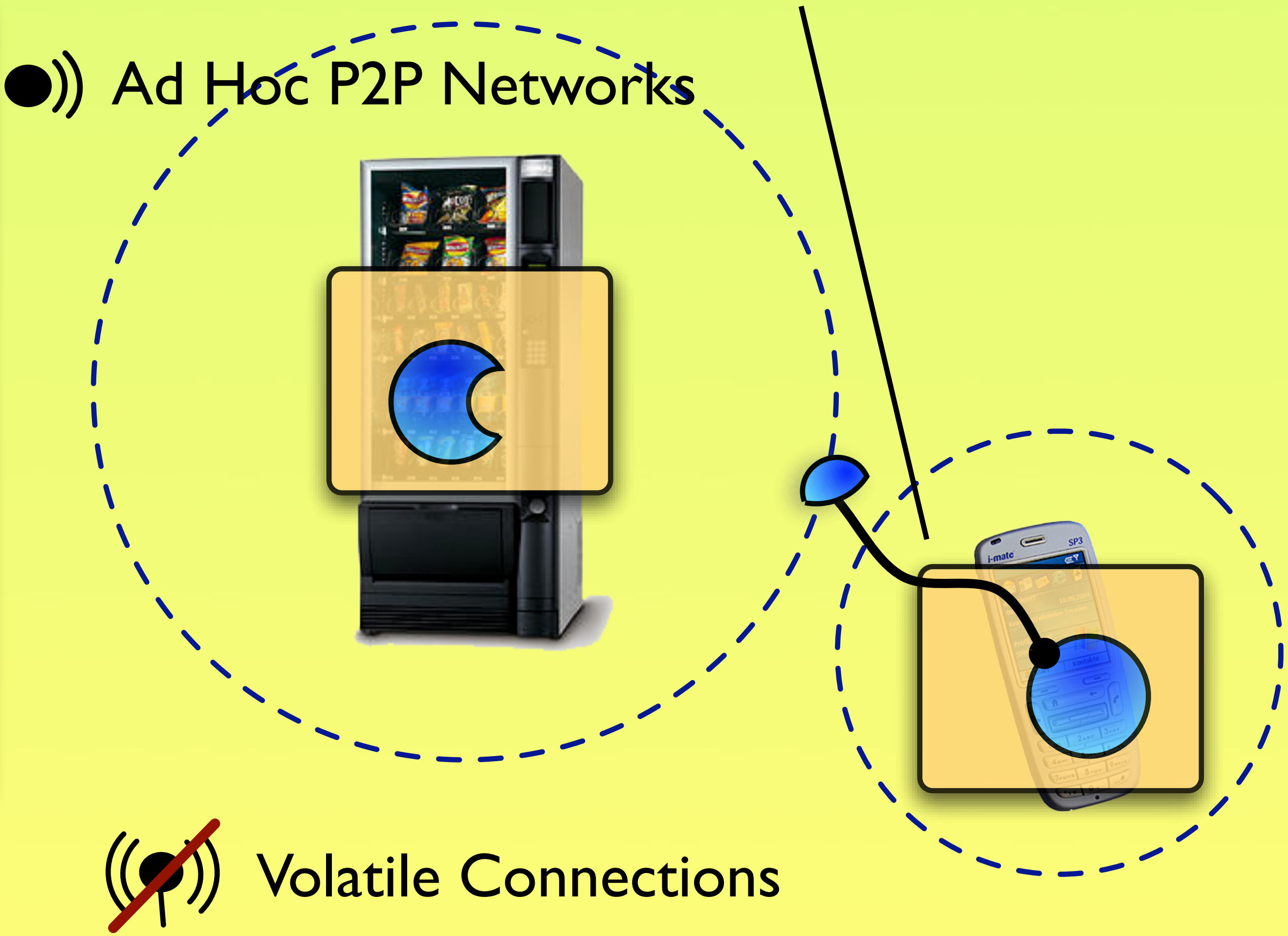
cellPhone#debit(price)

vendingMachine#getProduct(id)

Mobile Networks

Pervasive/Ubiquitous Computing

Ambient References
“remote actor references”



●)) Ad Hoc P2P Networks

Volatile Connections

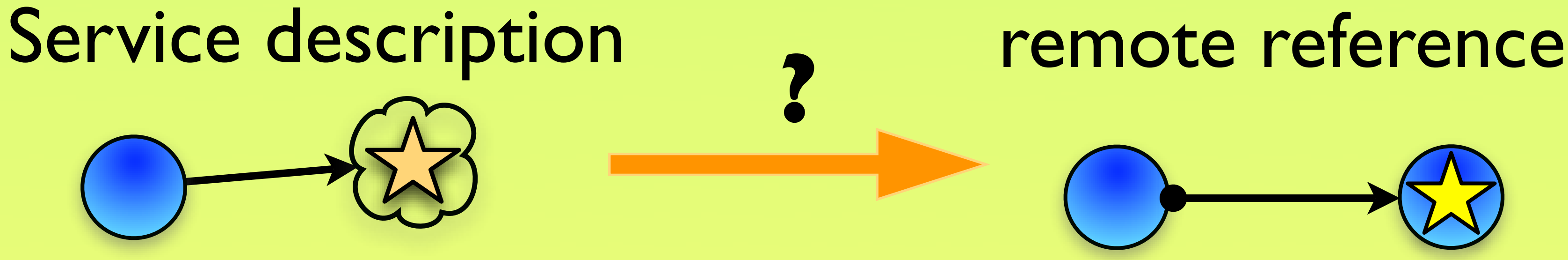
Provisionality

Provisionality

Resilience

Trans. Addressing

Group Comm.



Service description

?

remote reference

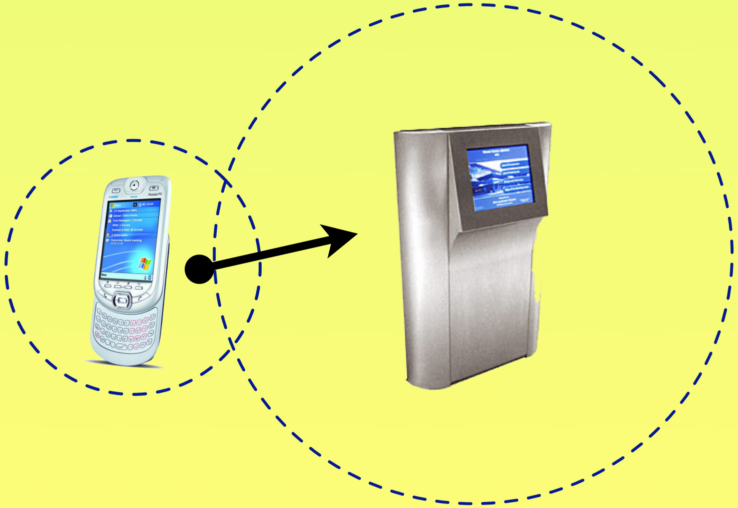
p2p discovery?

- ○ concurrency control
- ○ callbacks partition code
- ○ managing disconnections

```
...
Discovery.search(serviceDescription,
  new DiscoveryListener() {
    void foundService(Service s) {
      // use the service
    }
    void lostService(Service s) {
      // manage disconnection
    }
  });
```

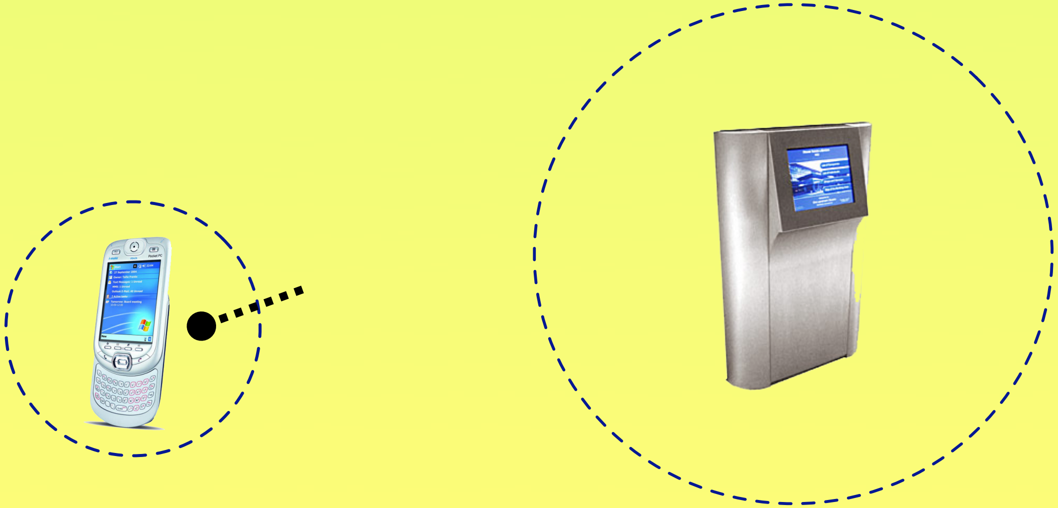
Resilience

- Temporary disconnections
  - should not break a remote reference
  - should not immediately raise exceptions
- Communication should resume upon reconnection



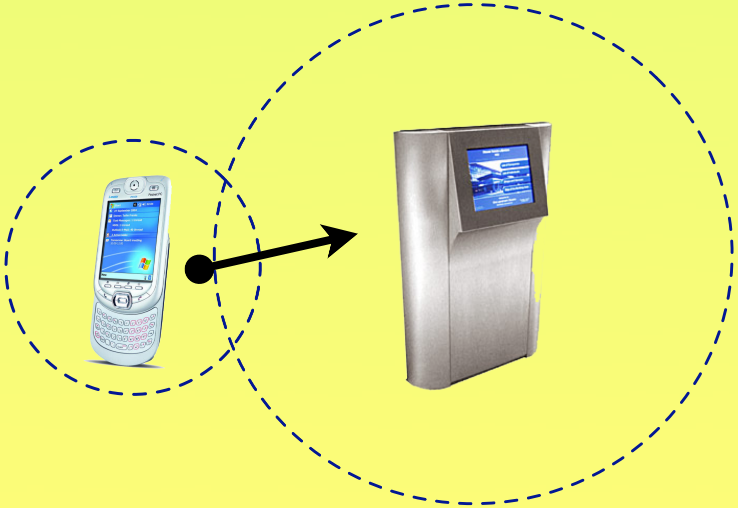
Resilience

- Temporary disconnections
  - should not break a remote reference
  - should not immediately raise exceptions
- Communication should resume upon reconnection



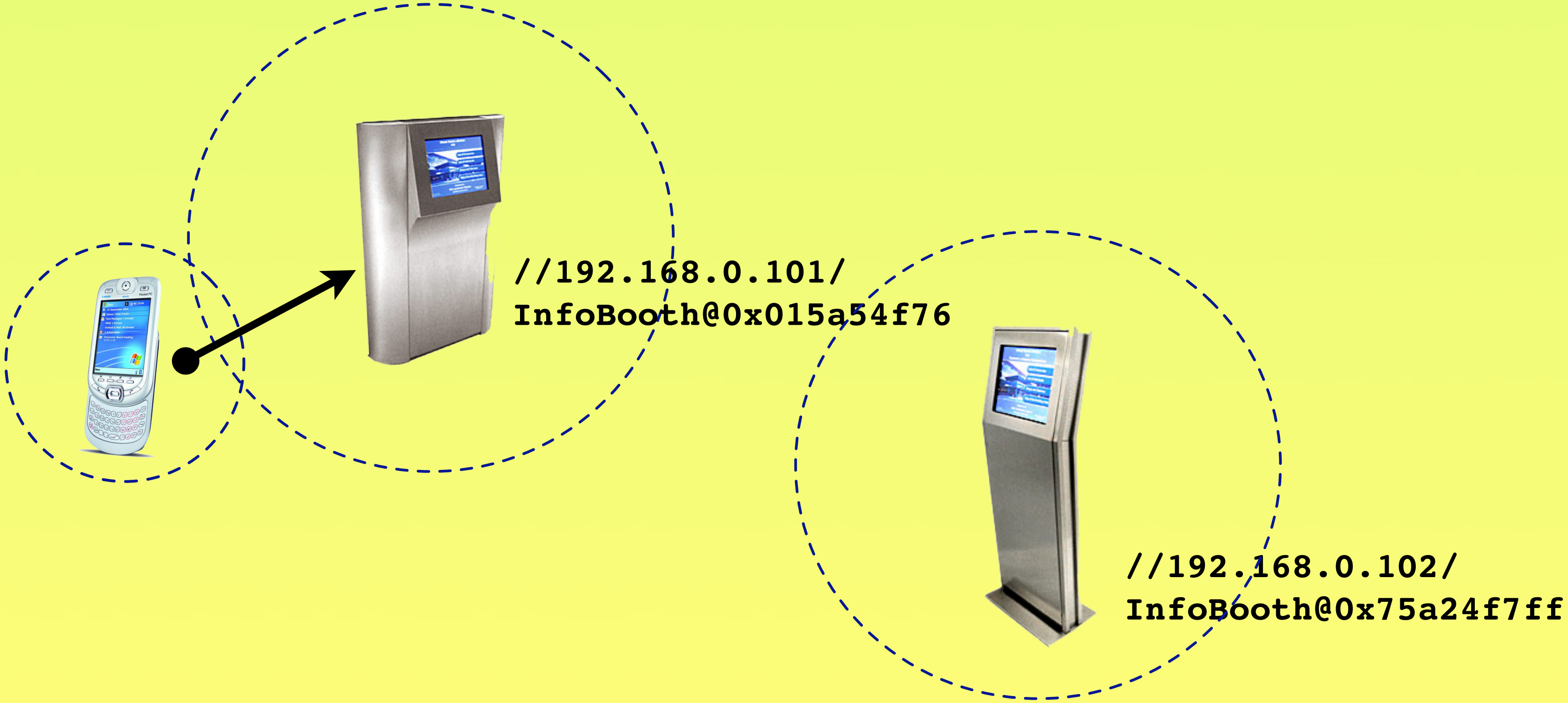
Resilience

- Temporary disconnections
  - should not break a remote reference
  - should not immediately raise exceptions
- Communication should resume upon reconnection



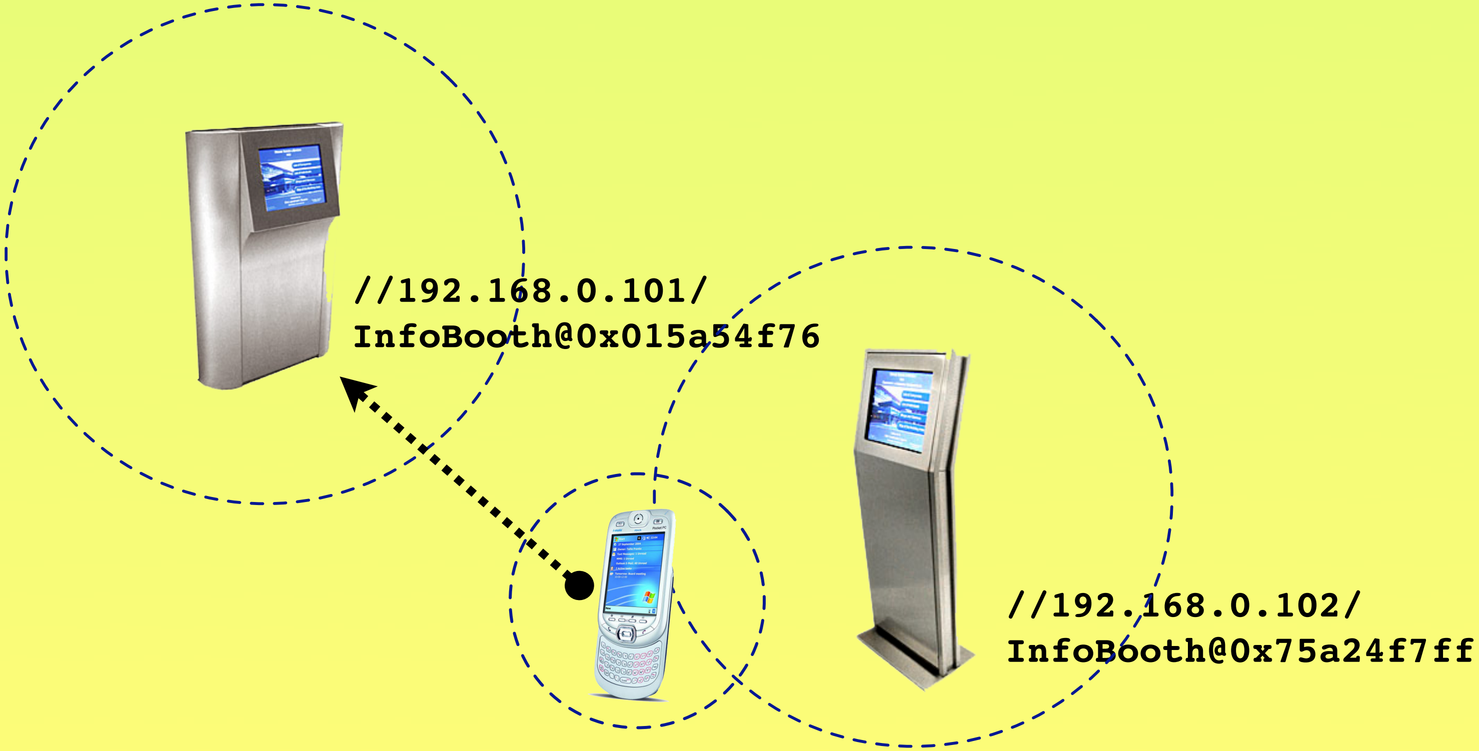
Transitory Addressing

- Remote references: UID-based, often device-dependent
- Too inflexible: cannot rebind



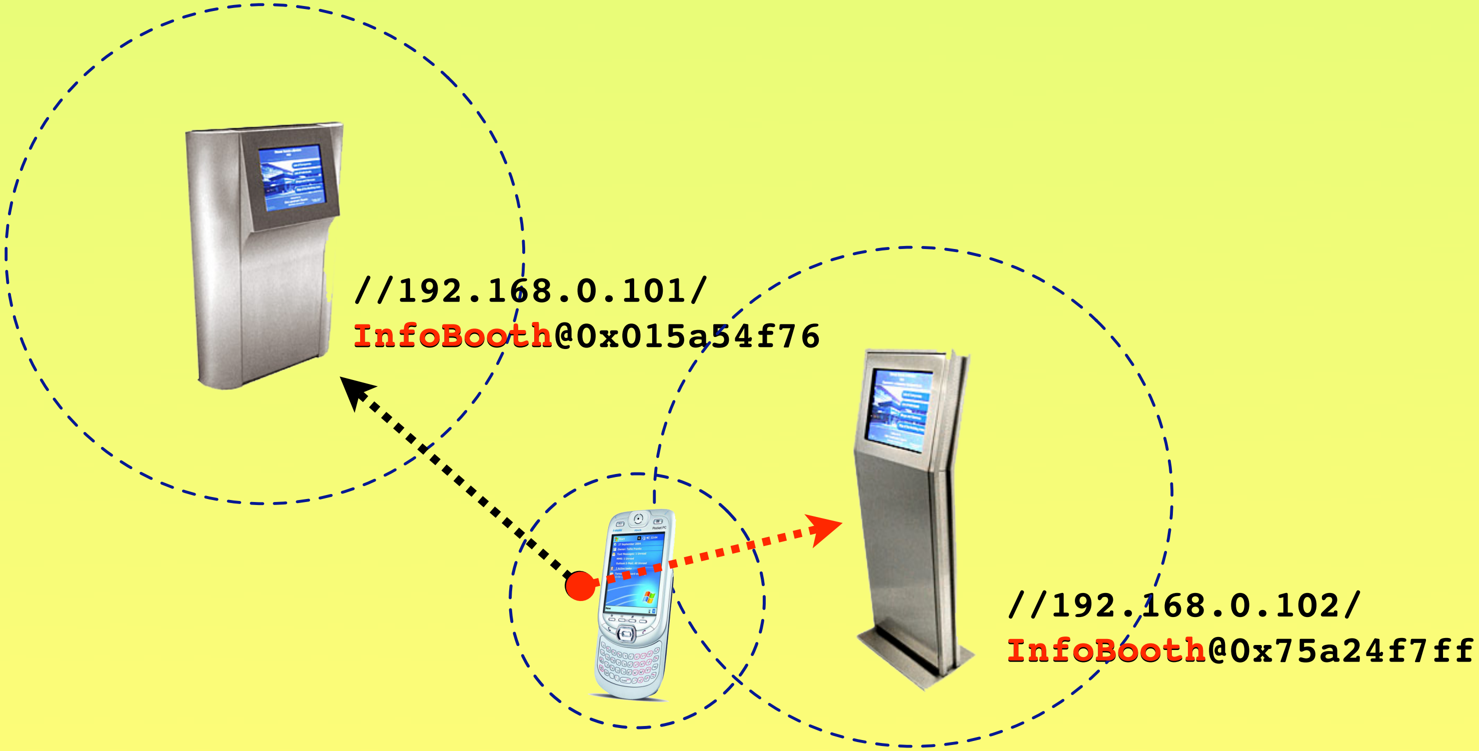
Transitory Addressing

- Remote references: UID-based, often device-dependent
- Too inflexible: cannot rebind



Transitory Addressing

- Remote references: UID-based, often device-dependent
- Too inflexible: cannot rebind

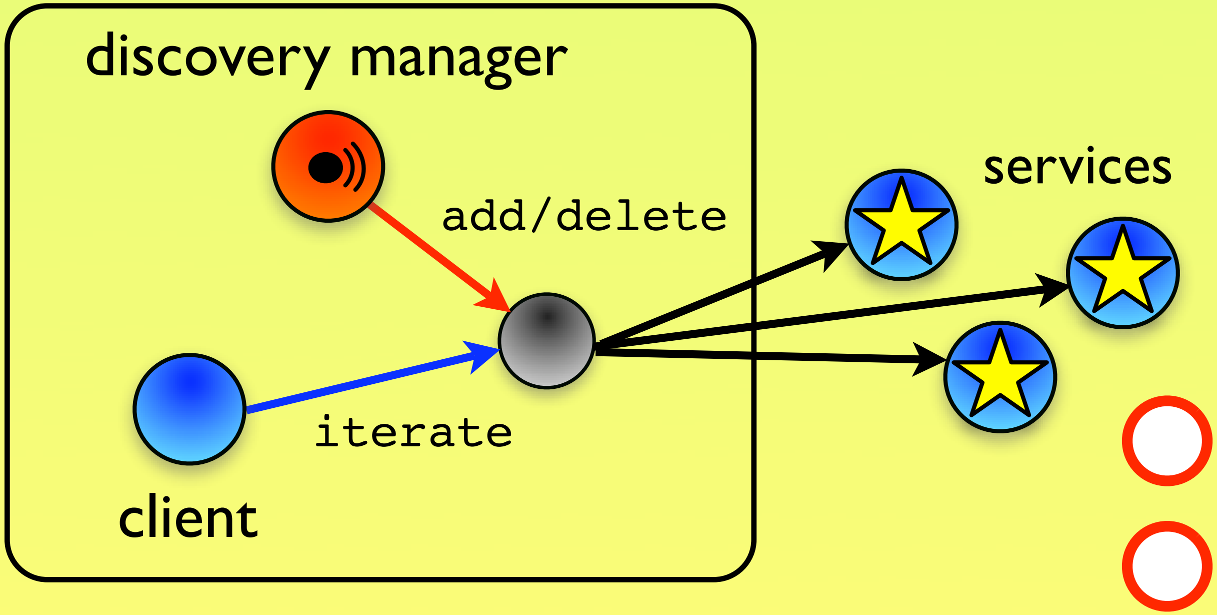


Group Communication

Provisionality
Resilience
Trans. Addressing
Group Comm.

- Abstract from multitude of devices
- Ad hoc ‘proximate’ groups

collections?



○ concurrency control

○ boilerplate iteration code

Problem Statement

Provisionality

Resilience

Transitory Addressing

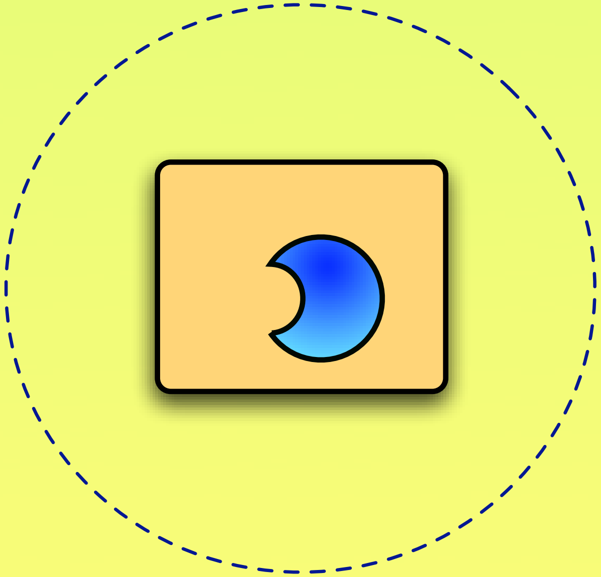
Group Communication

- Standard remote object references fail to meet these requirements
- Need for **dedicated** referencing abstractions for mobile networks

Computational Model

- Services are ‘public’ **actors** advertising themselves via *service types*

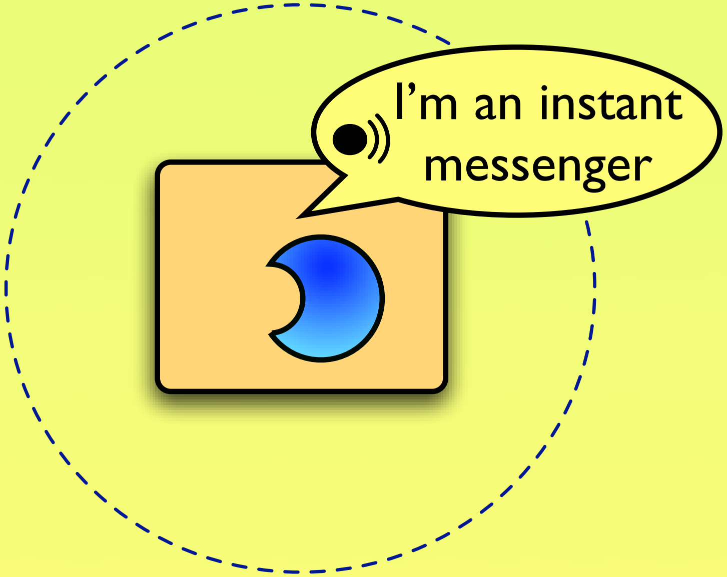
```
servicetype InstantMessenger;

actor {
  provide(InstantMessenger);
  method talk(text) { ... };
}
```



Computational Model

- Services are ‘public’ **actors** advertising themselves via *service types*

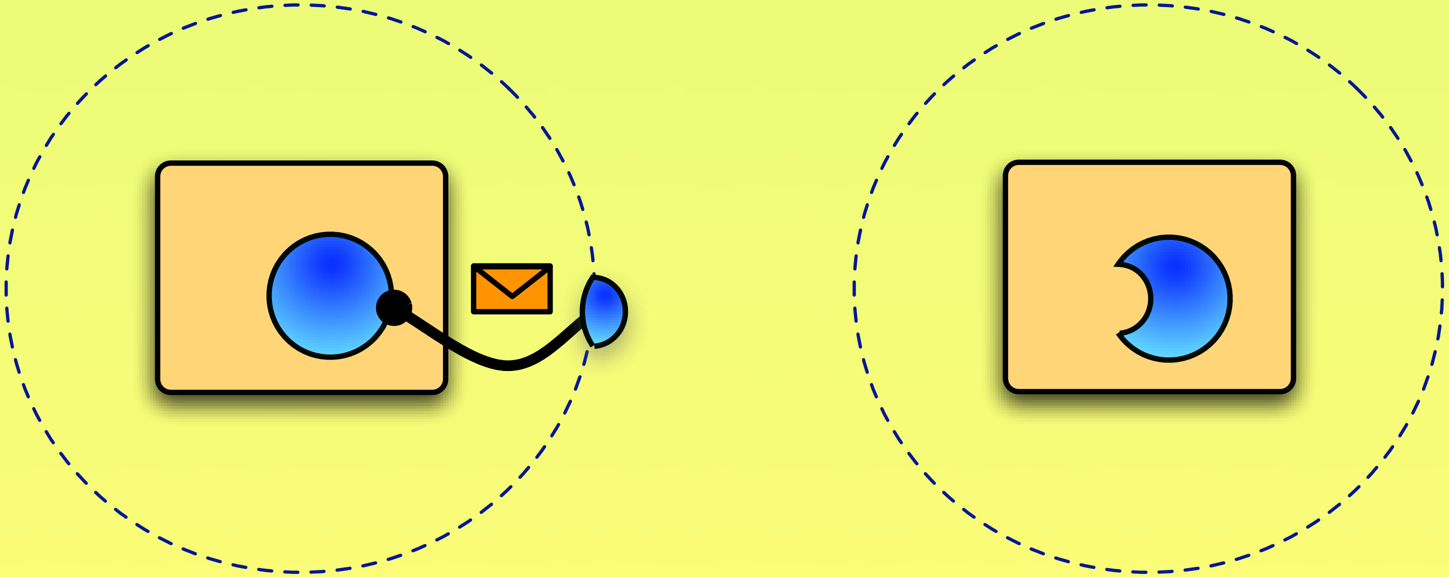
```
servicetype InstantMessenger;

actor {
  provide(InstantMessenger);
  method talk(text) { ... };
}
```



Ambient References

- Two states: **bound** or **unbound**
- Binds to proximate matching services

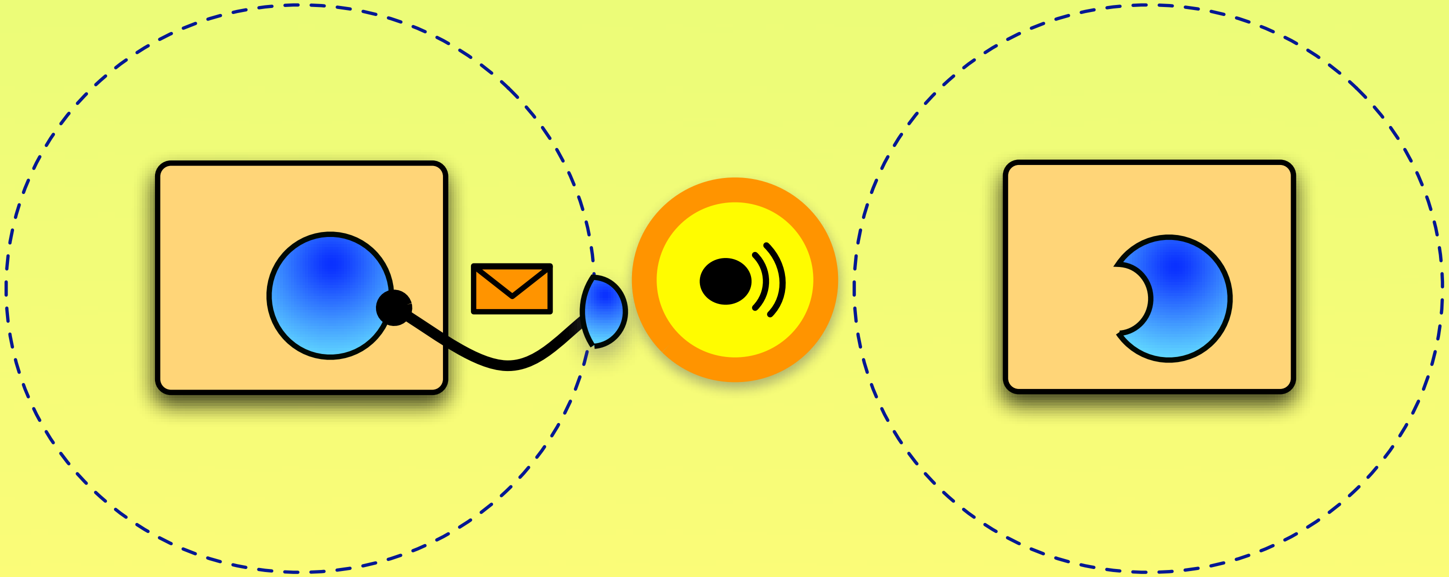
```
aMessenger = ambient InstantMessenger
aMessenger#talk(“Hello”)
```



Ambient References

- Two states: **bound** or **unbound**
- Binds to proximate matching services

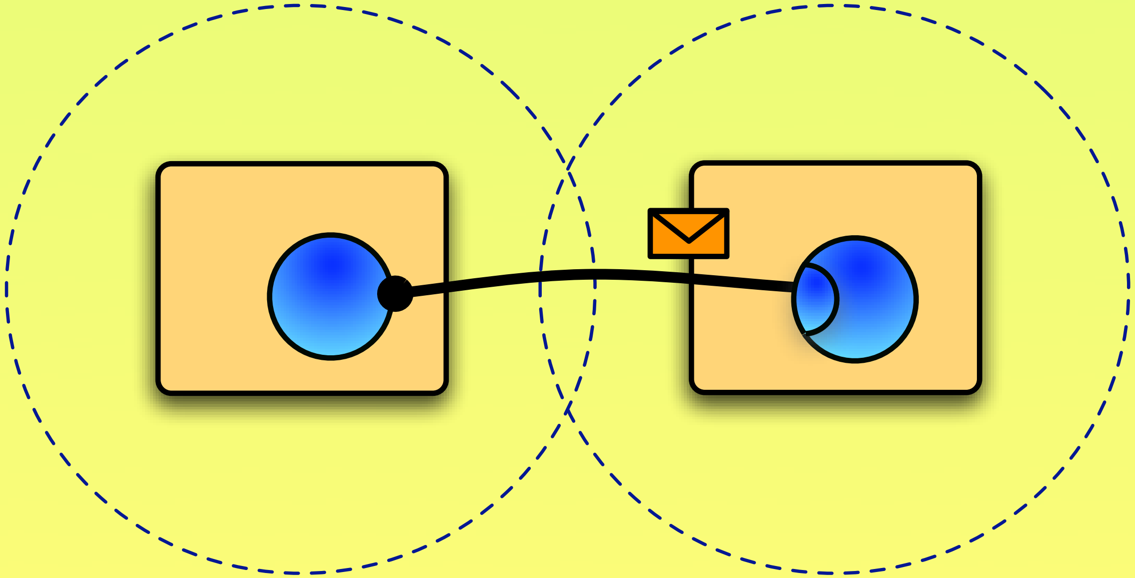
```
aMessenger = ambient InstantMessenger
aMessenger#talk(“Hello”)
```



Ambient References

- Two states: **bound** or **unbound**
- Binds to proximate matching services

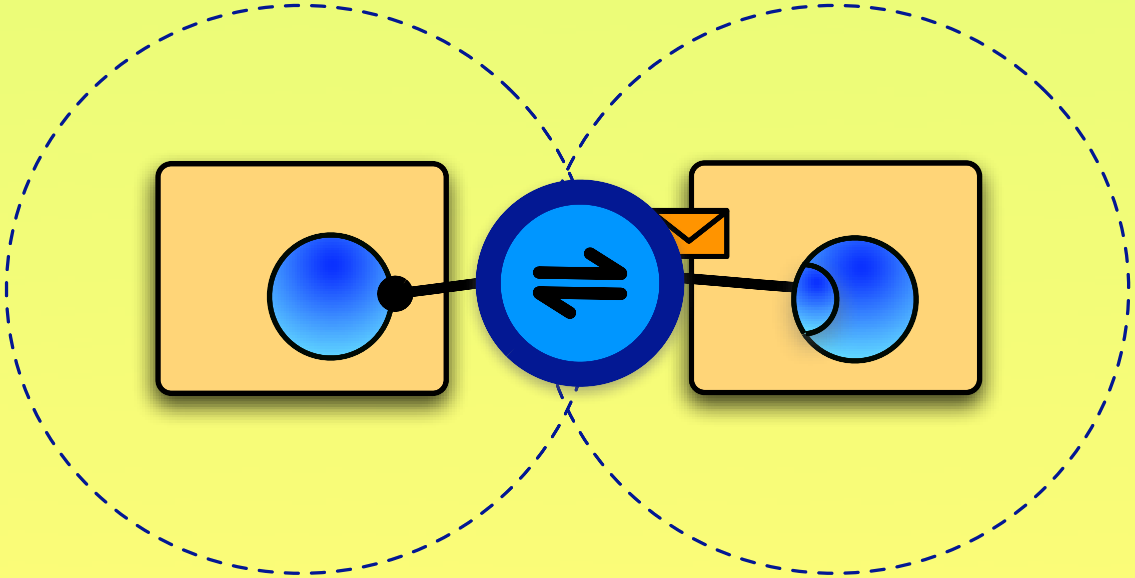
```
aMessenger = ambient InstantMessenger
aMessenger#talk(“Hello”)
```



Ambient References

- Two states: **bound** or **unbound**
- Binds to proximate matching services

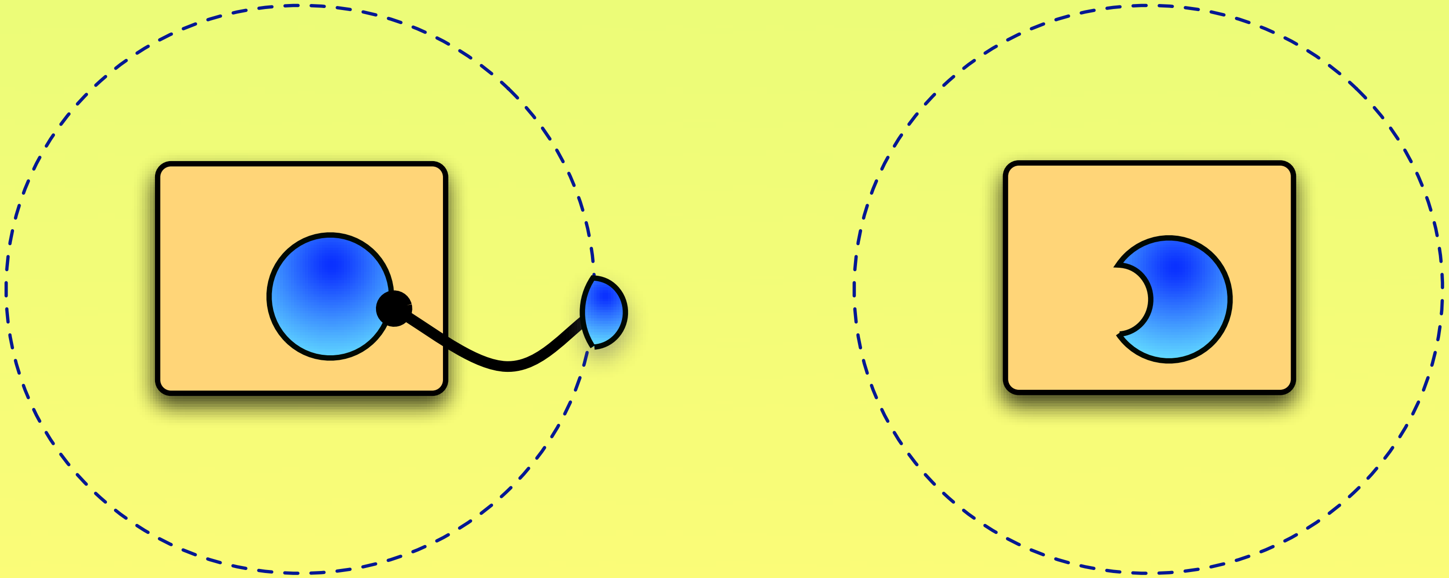
```
aMessenger = ambient InstantMessenger
aMessenger#talk("Hello")
```



Ambient References

- Two states: **bound** or **unbound**
- Binds to proximate matching services

```
aMessenger = ambient InstantMessenger
aMessenger#talk("Hello")
```

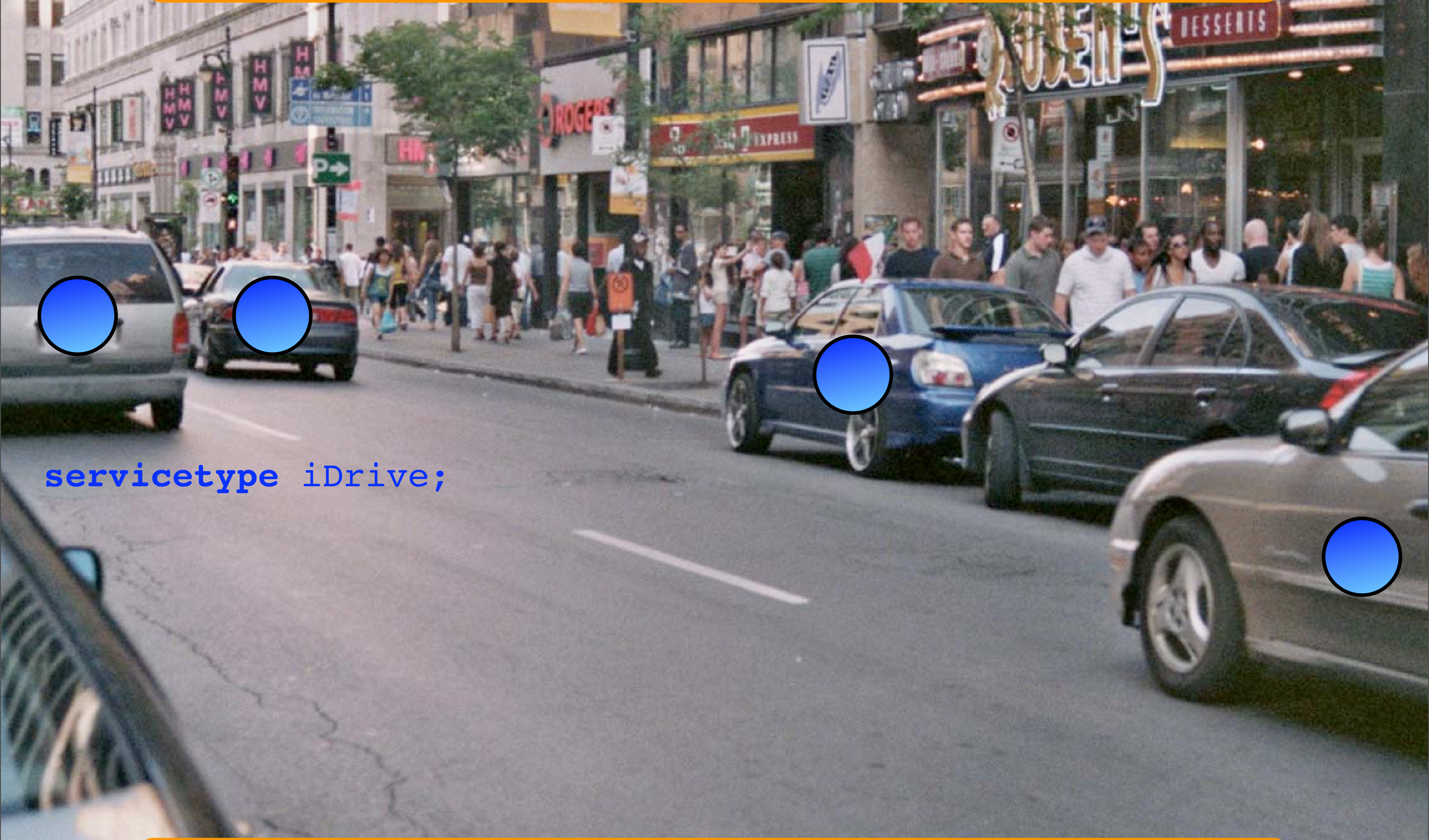


Design Dimensions

- Design **family** of remote references
- each suitable for different kind of collaboration
- Three properties:
  - Scope of binding
  - Elasticity
  - Cardinality

Scope of binding

Scope of binding



```
servicetype iDrive;
```

Scope of binding

```
servicetype iSign;
```

```
servicetype iDrive;
```

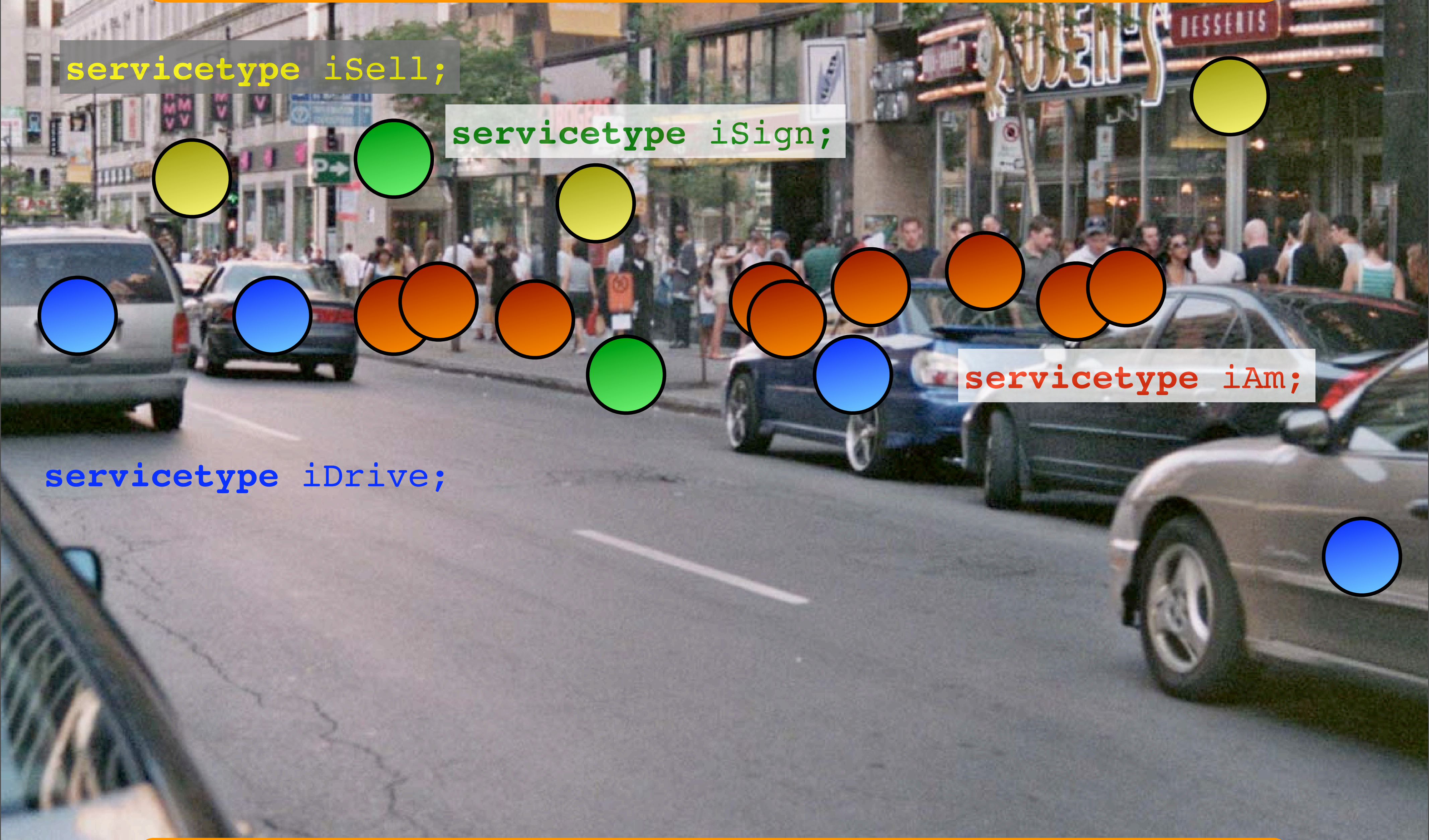
Scope of binding

```
servicetype iSell;
```

```
servicetype iSign;
```

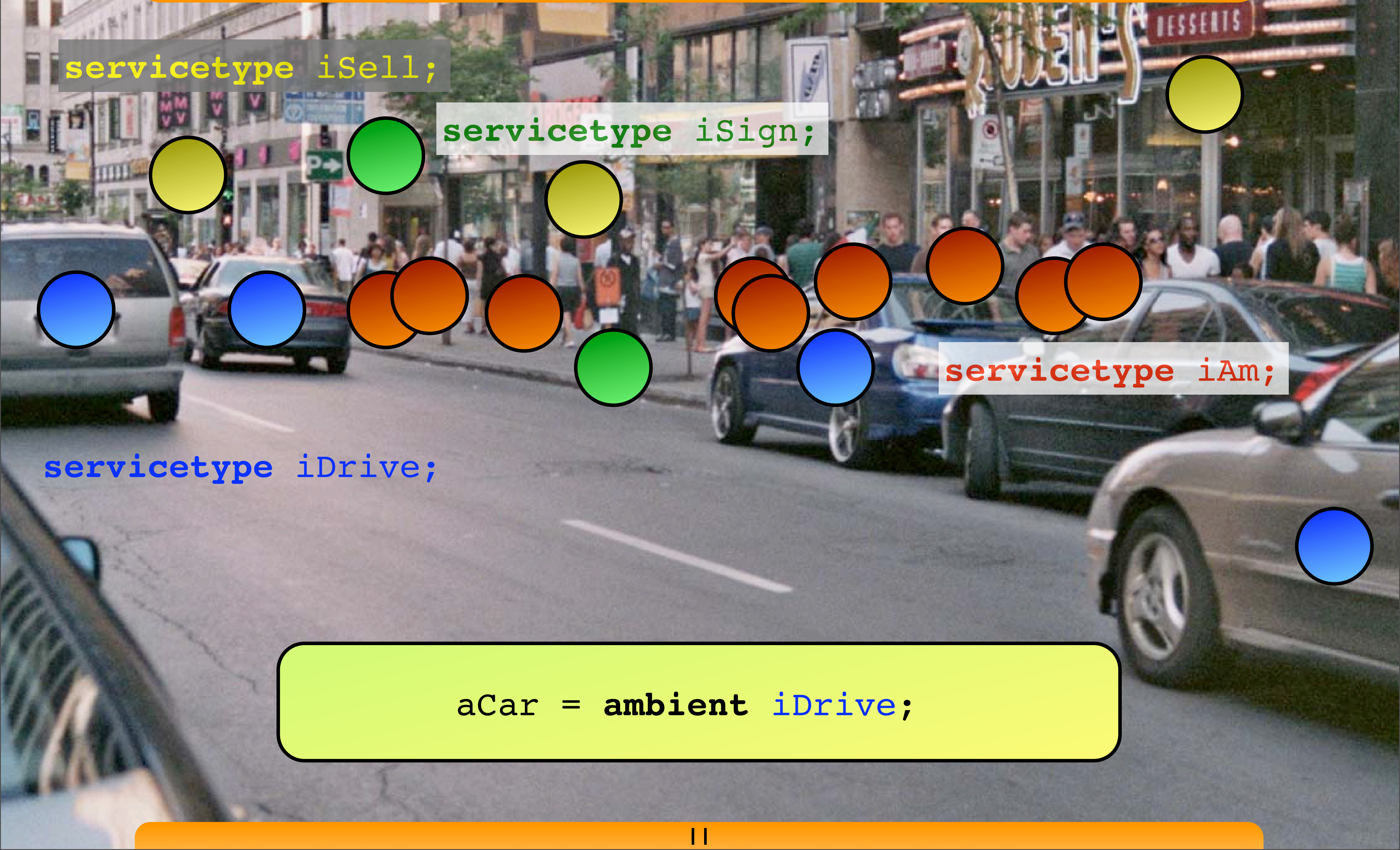
```
servicetype iDrive;
```

Scope of binding



```
servicetype iSell;
servicetype iSign;
servicetype iAm;
servicetype iDrive;
```

Scope of binding



```
servicetype iSell;
```

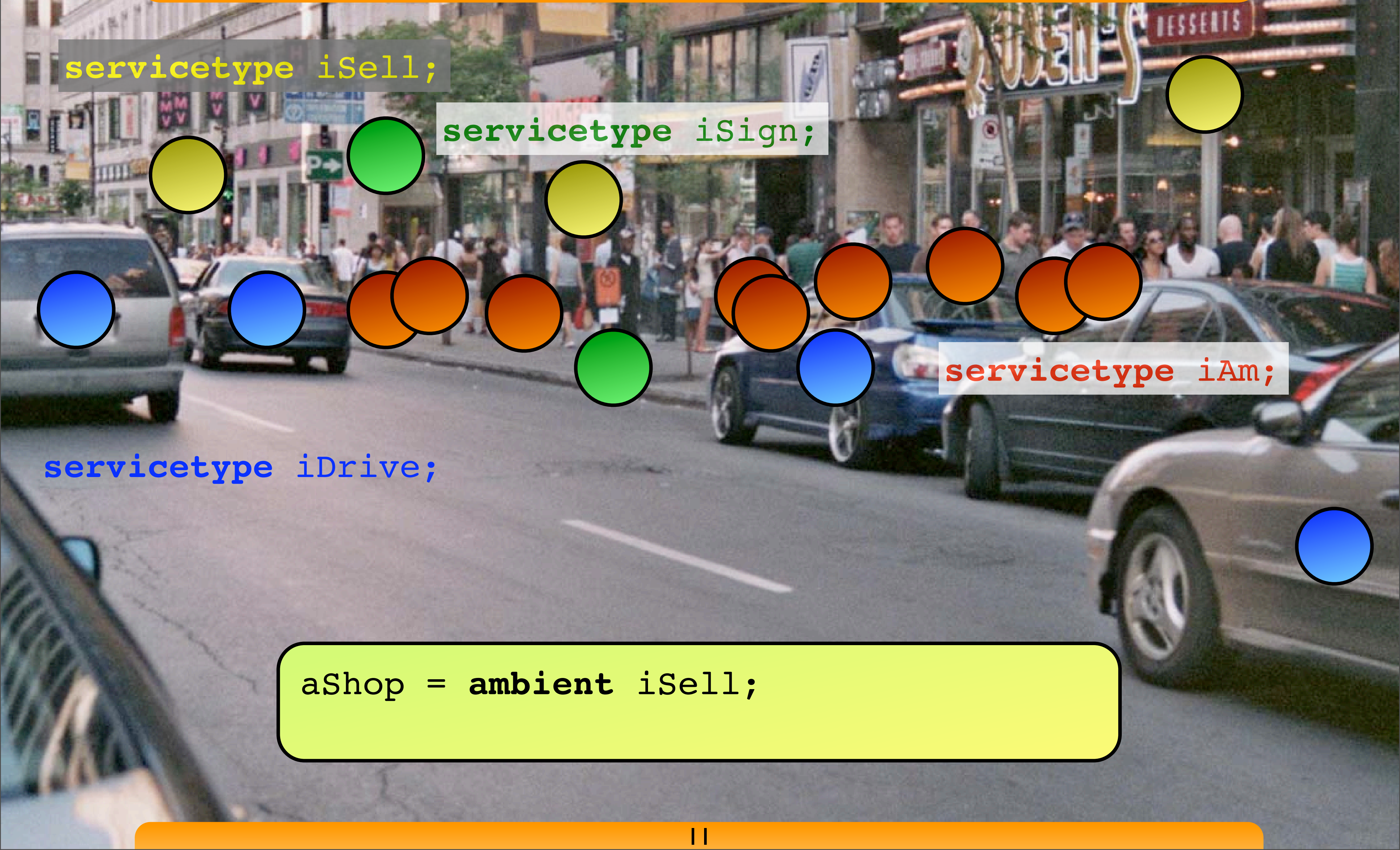
```
servicetype iSign;
```

```
servicetype iAm;
```

```
servicetype iDrive;
```

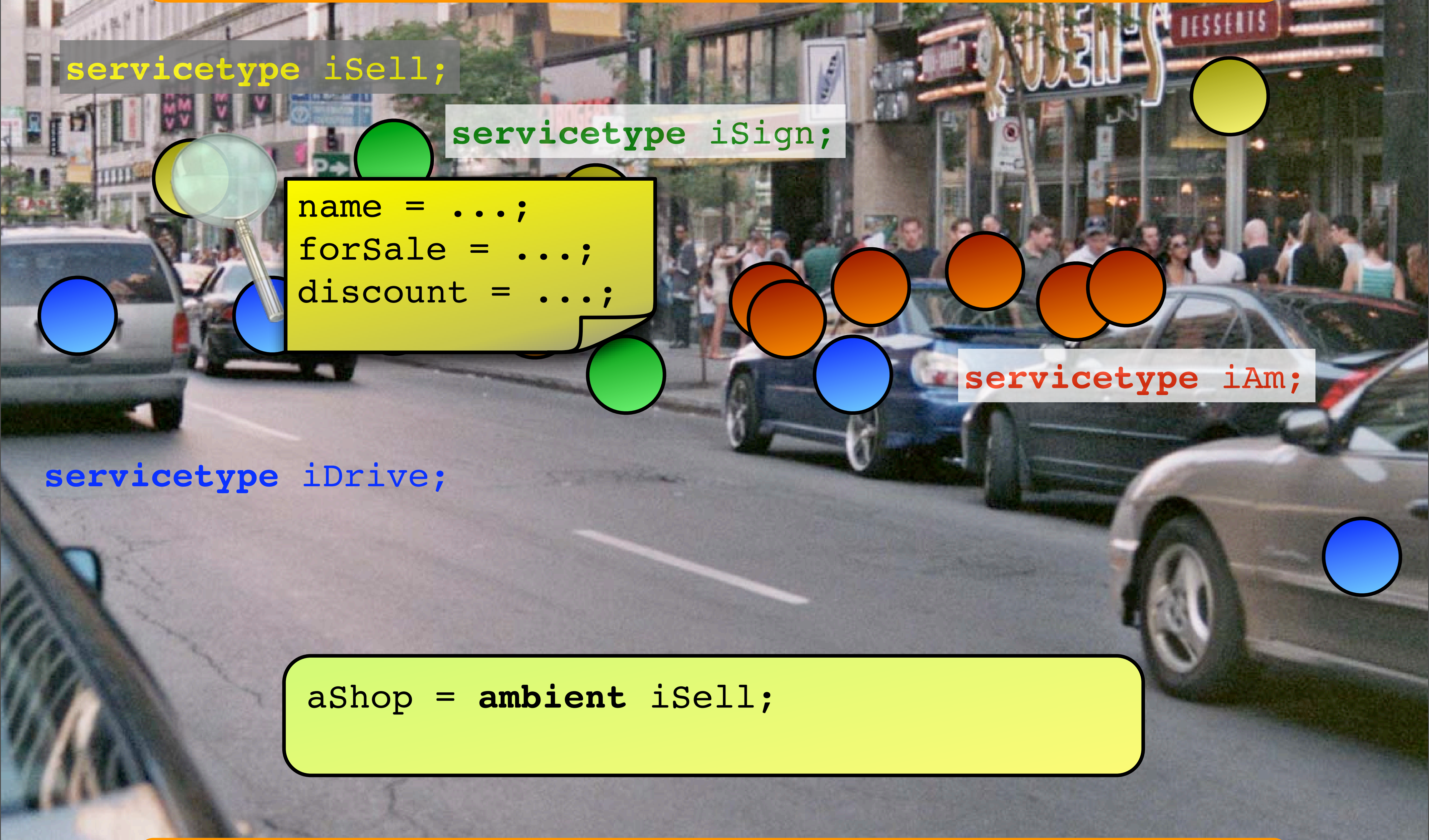
```
aCar = ambient iDrive;
```

Scope of binding



```
aShop = ambient iSell;
```

Scope of binding



```
servicetype iSell;
```

```
servicetype iSign;
```

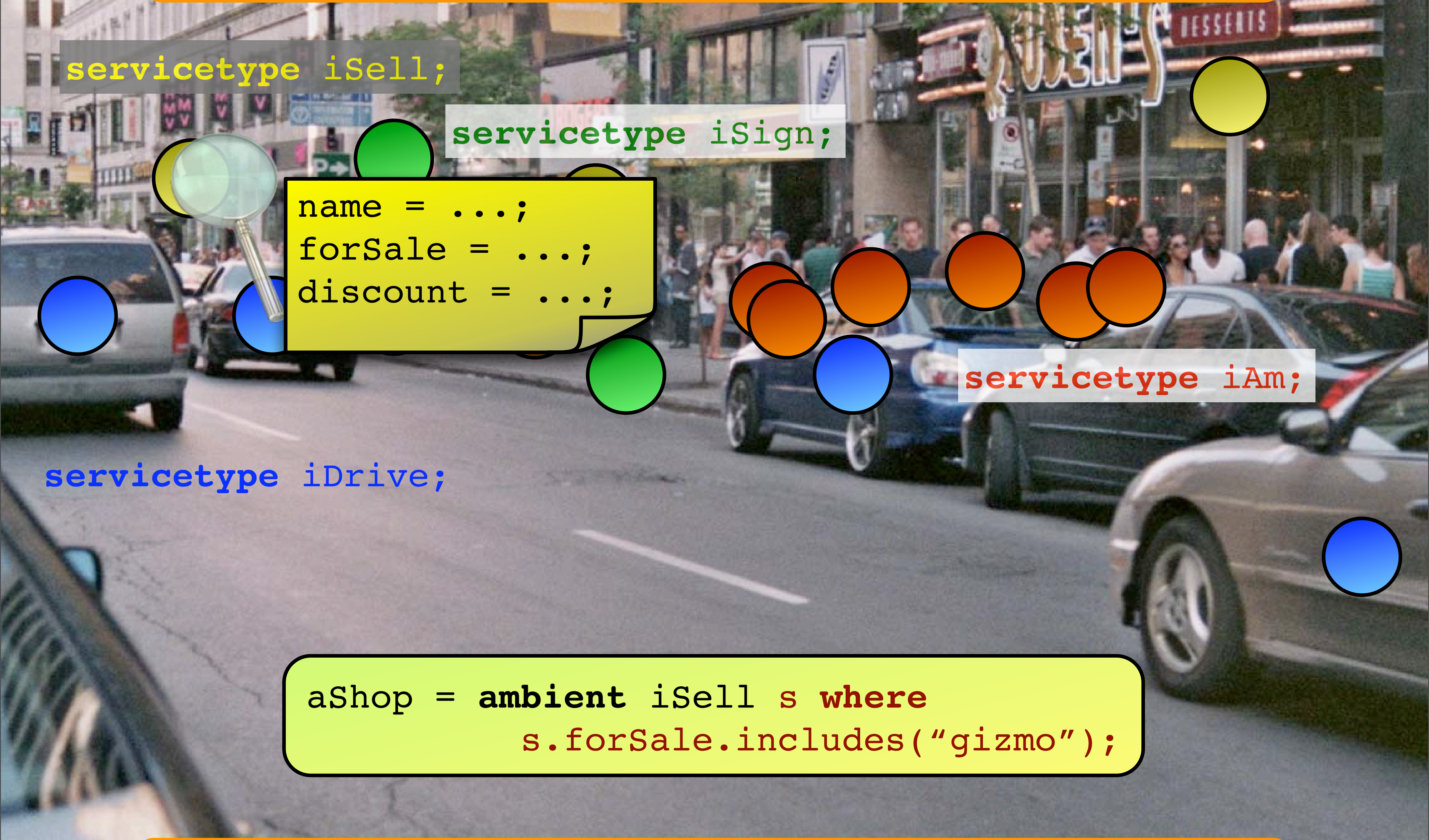
```
name = ...;
forSale = ...;
discount = ...;
```

```
servicetype iAm;
```

```
servicetype iDrive;
```

```
aShop = ambient iSell;
```

Scope of binding



```
servicetype iSell;
```

```
servicetype iSign;
```

```
name = ...;
forSale = ...;
discount = ...;
```

```
servicetype iAm;
```

```
servicetype iDrive;
```

```
aShop = ambient iSell s where
          s.forSale.includes("gizmo");
```

Elasticity

```
servicetype iSell;
```

```
servicetype iSign;
```

```
servicetype iAm;
```

```
servicetype iDrive;
```

Fragile

Elastic

Sturdy

Elasticity

```
servicetype iSell;
```

```
servicetype iSign;
```

```
servicetype iAm;
```

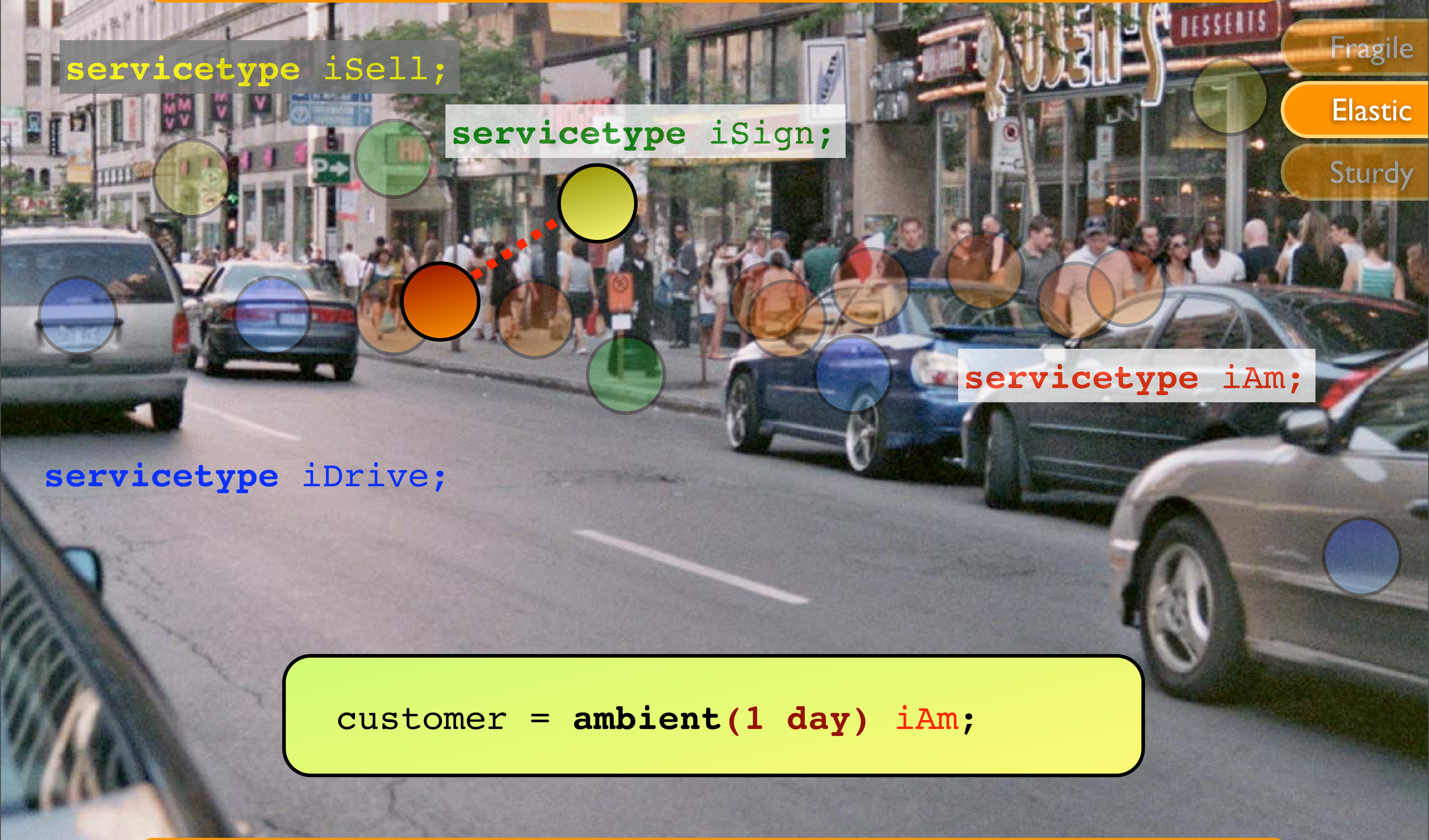
```
servicetype iDrive;
```

Fragile

Elastic

Sturdy

Elasticity



Fragile

Elastic

Sturdy

```
servicetype iSell;
```

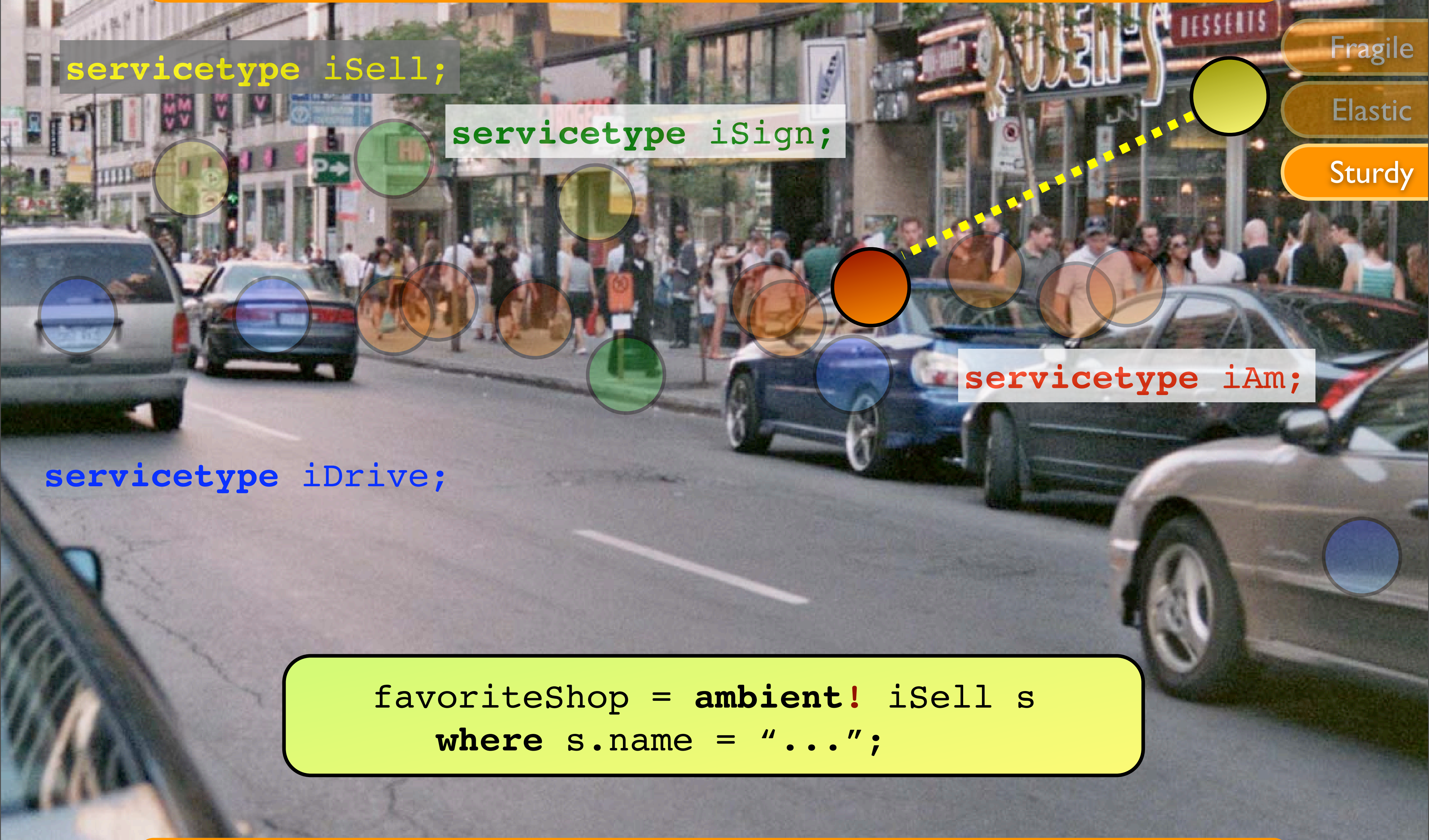
```
servicetype iSign;
```

```
servicetype iAm;
```

```
servicetype iDrive;
```

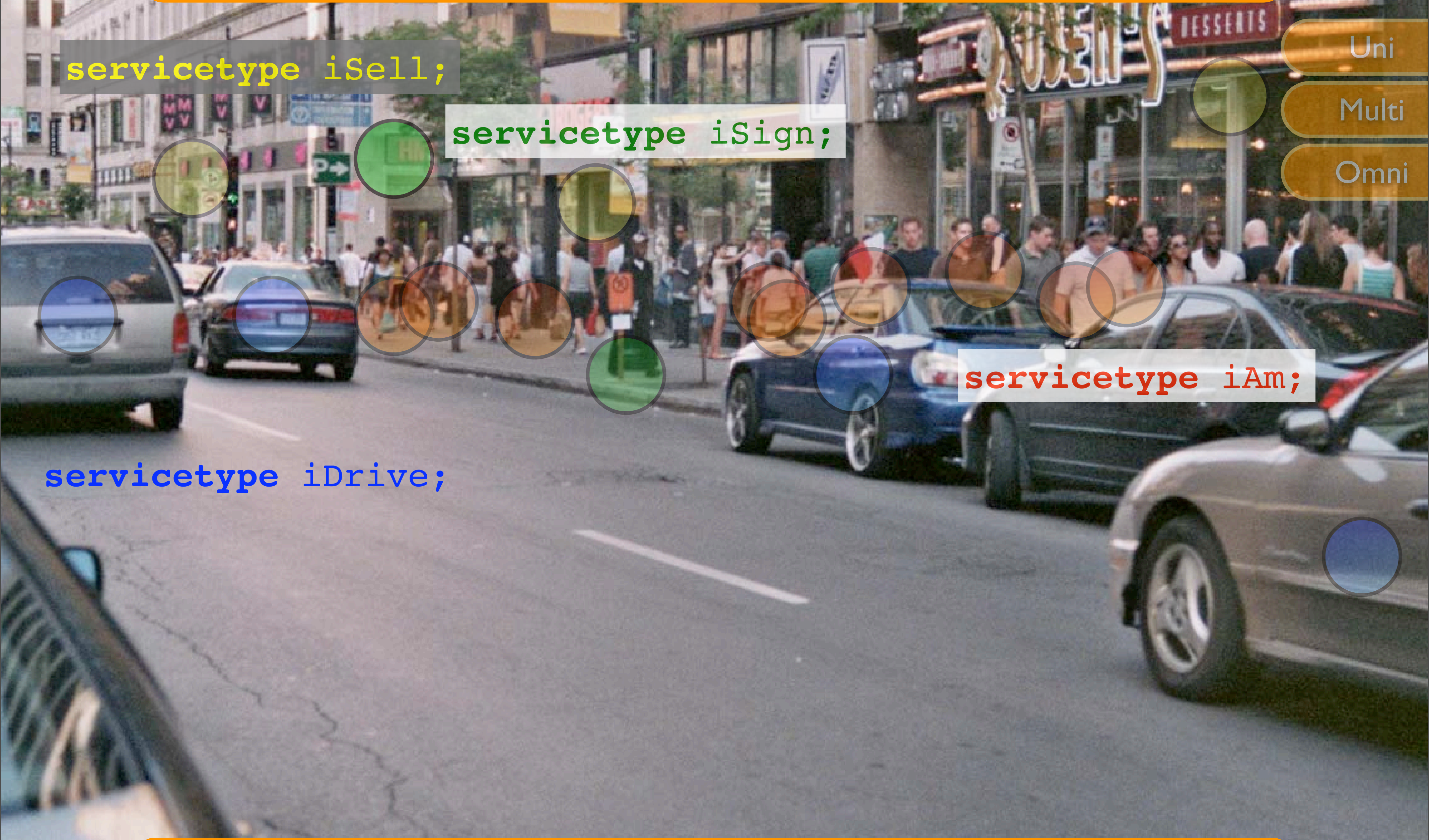
```
customer = ambient(1 day) iAm;
```

Elasticity



```
favoriteShop = ambient! iSell s
    where s.name = "...";
```

Cardinality



```
servicetype iSell;
```

```
servicetype iSign;
```

```
servicetype iAm;
```

```
servicetype iDrive;
```

Uni

Multi

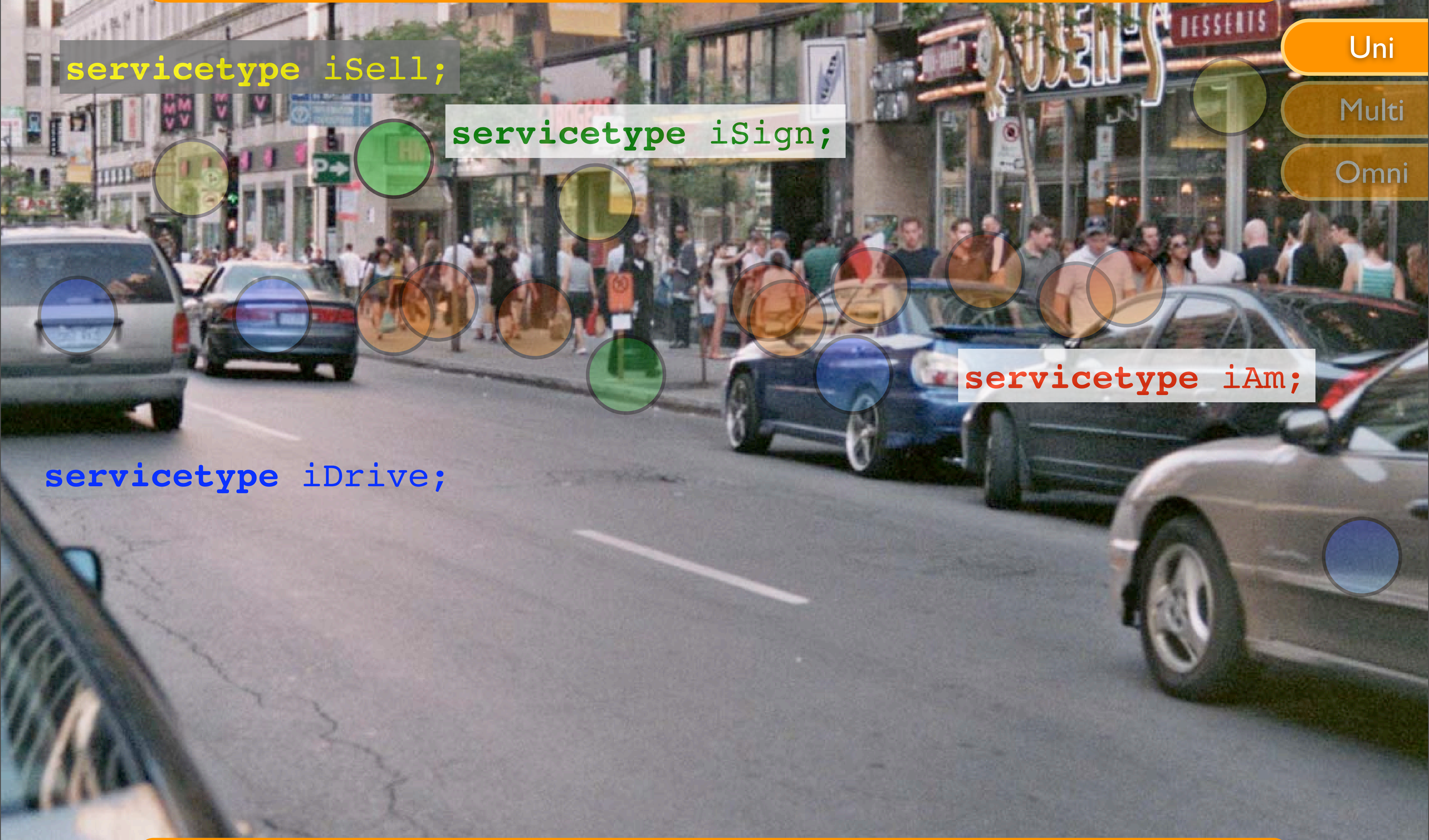
Omni

Cardinality

Uni

Multi

Omni



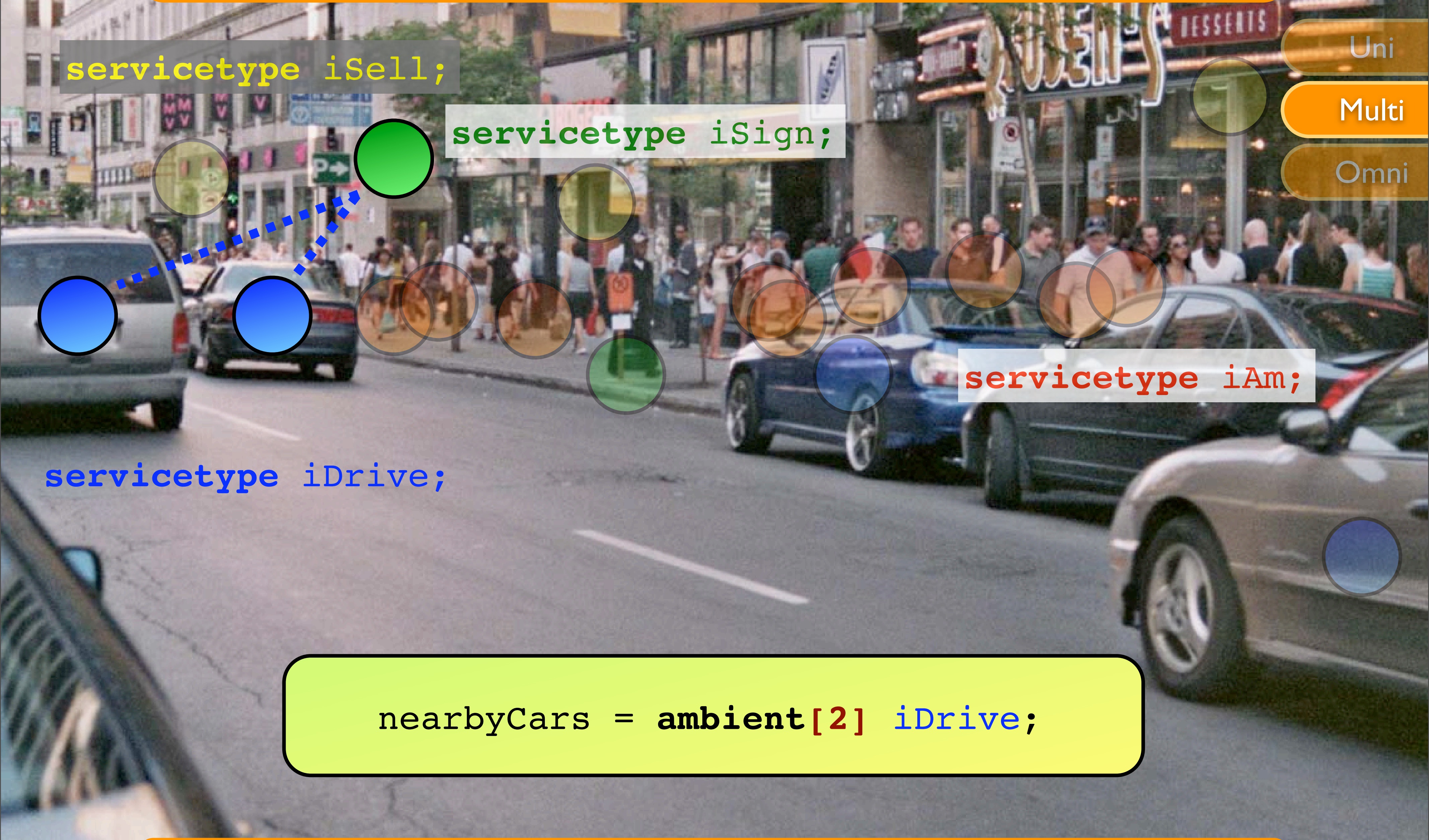
```
servicetype iSell;
```

```
servicetype iSign;
```

```
servicetype iAm;
```

```
servicetype iDrive;
```

Cardinality



```
servicetype iSell;
servicetype iSign;
servicetype iAm;
servicetype iDrive;
```

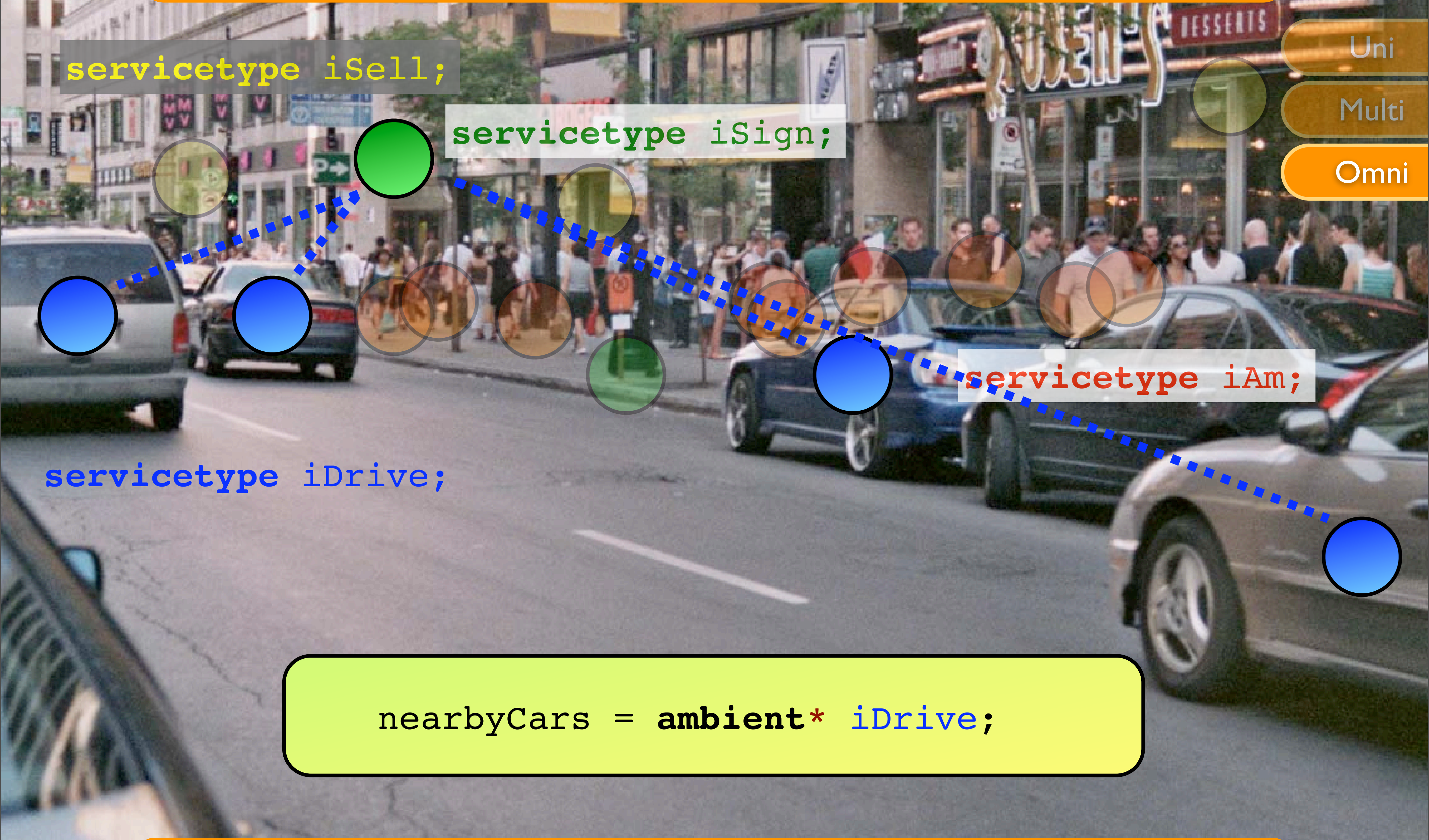
Uni

Multi

Omni

```
nearbyCars = ambient[2] iDrive;
```

Cardinality



```
servicetype iSell;
servicetype iSign;
servicetype iAm;
servicetype iDrive;
```

Uni

Multi

Omni

```
nearbyCars = ambient* iDrive;
```

Taxonomy

Scope of binding			
Elasticity x Cardinality	Fragile	Elastic	Sturdy
Uni	`ambient S;`	`ambient(e) S;`	`ambient! S;`
Multi	`ambient[n] S;`	`ambient(e)[n] S;`	`ambient![n] S;`
Omni	`ambient* S;`	`ambient(e)* S;`	`ambient!* S;`

Requirements revisited

Provisionality

Resilience

Transitory Addressing

Group Communication

Requirements revisited

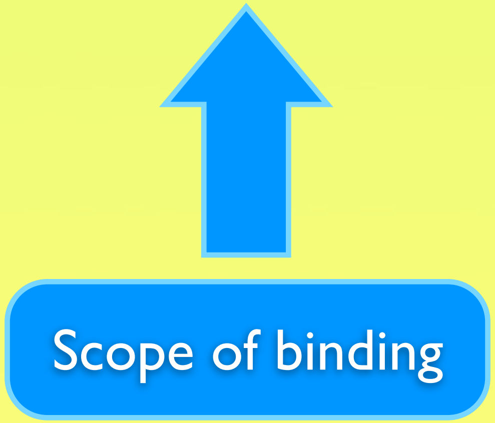
Provisionality

Resilience

Transitory Addressing

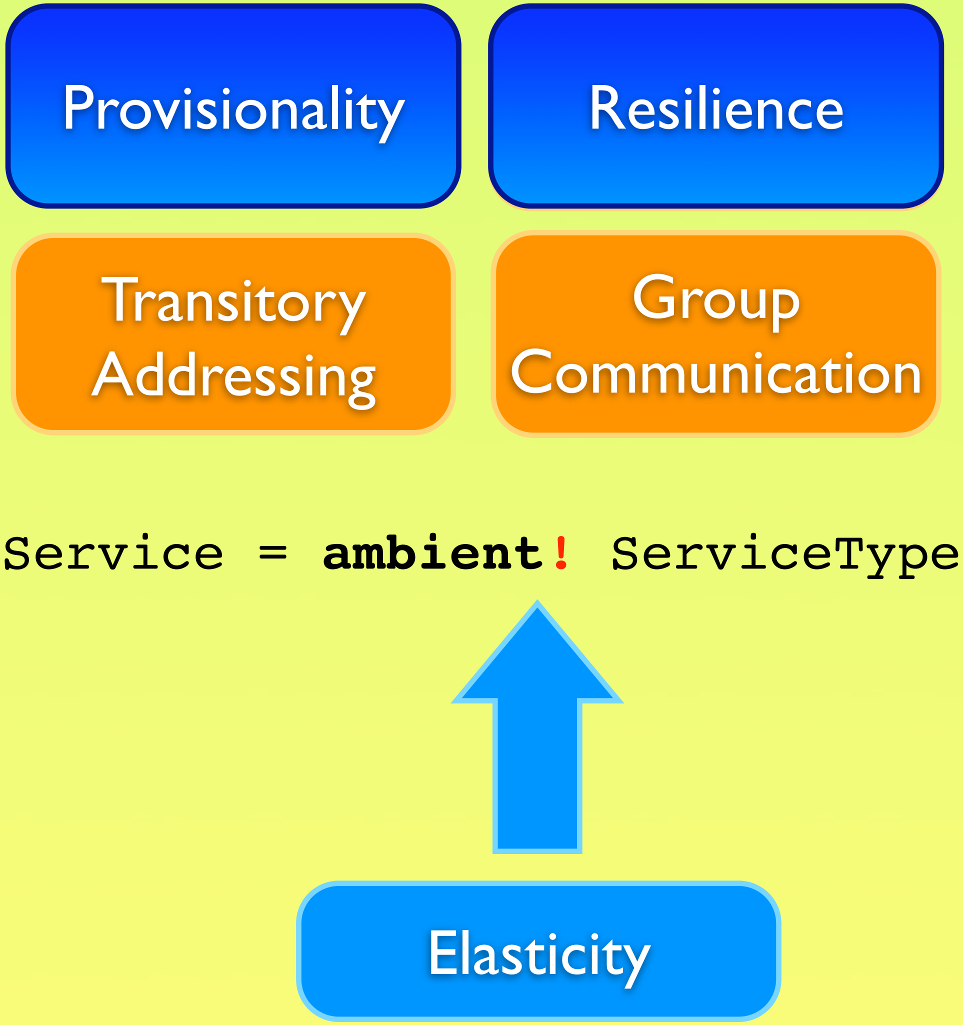
Group Communication

```
aService = ambient ServiceType
```



Scope of binding

Requirements revisited



Provisionality

Resilience

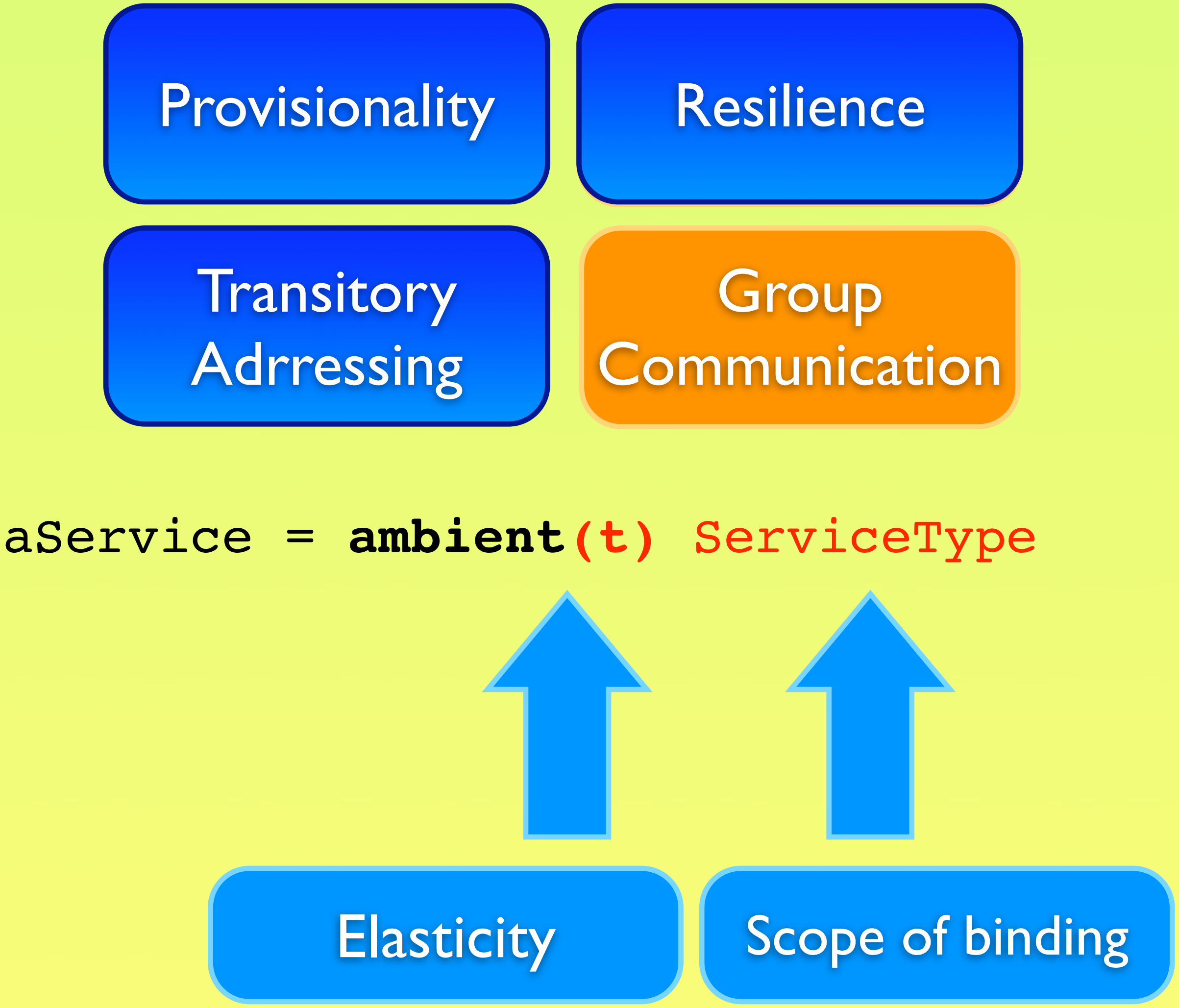
Transitory Addressing

Group Communication

```
aService = ambient! ServiceType
```

Elasticity

Requirements revisited



Provisionality

Resilience

Transitory Adrressing

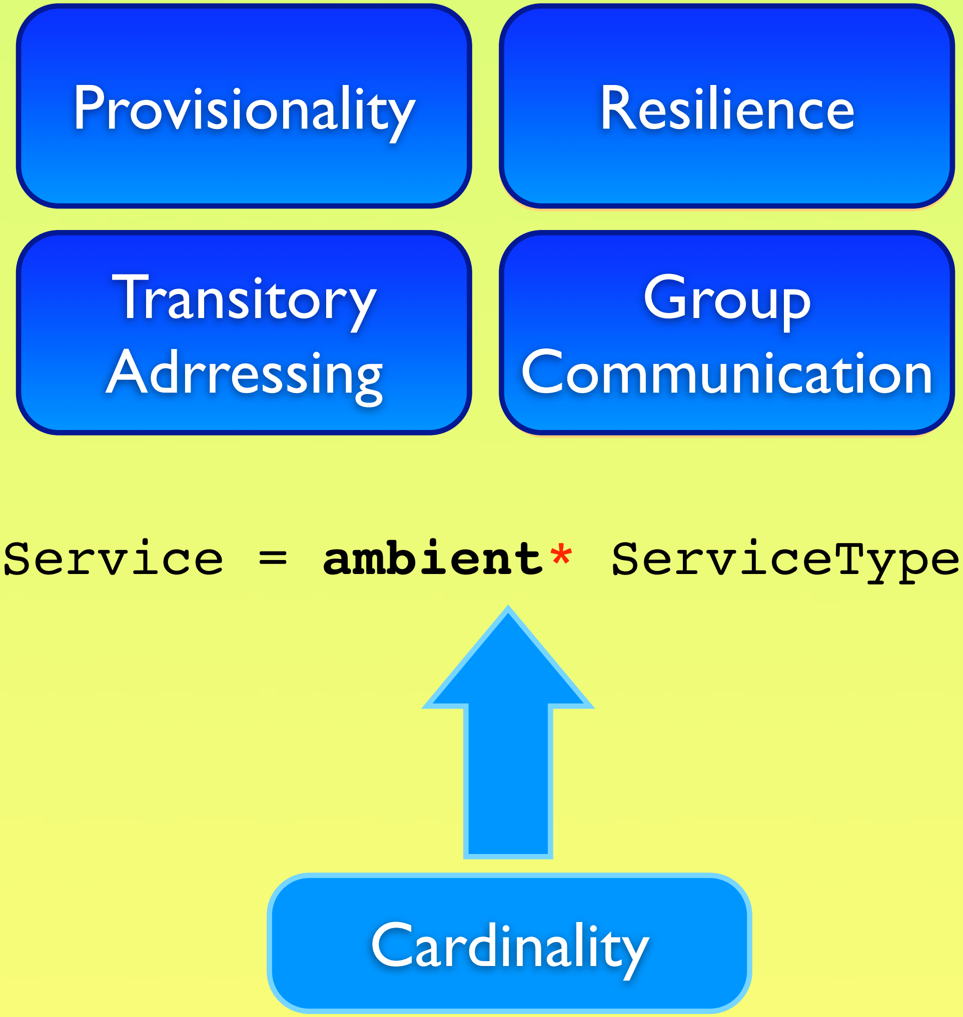
Group Communication

```
aService = ambient(t) ServiceType
```

Elasticity

Scope of binding

Requirements revisited



Provisionality

Resilience

Transitory Adrressing

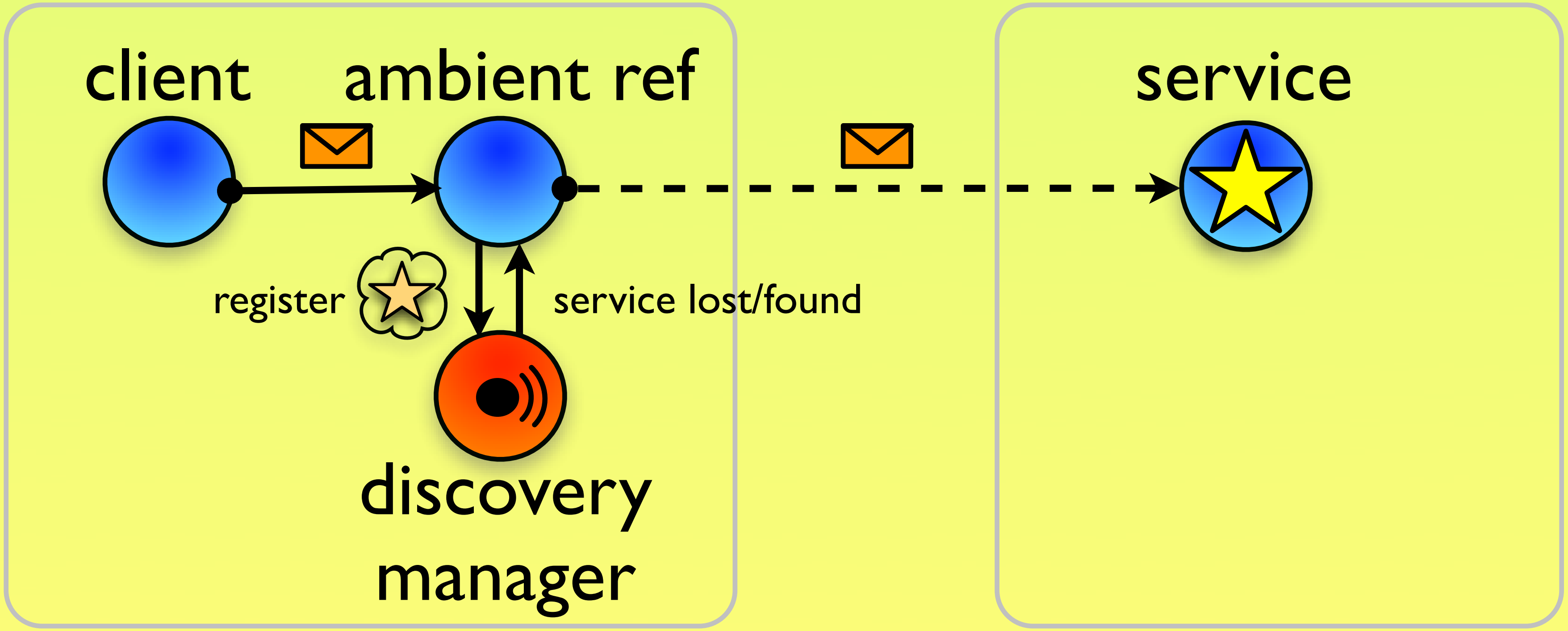
Group Communication

```
aService = ambient* ServiceType
```

Cardinality

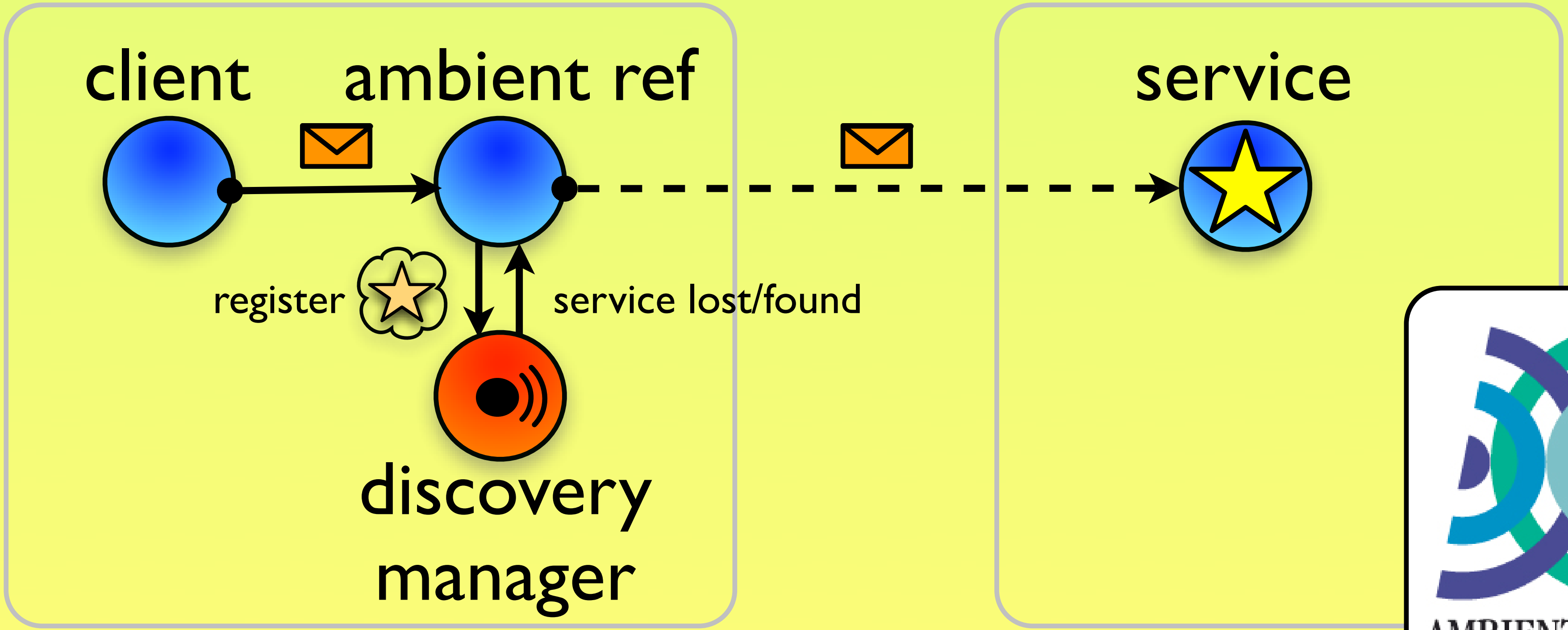
Implementation

- Local **proxy** for remote service
- Performs discovery on behalf of its client



Implementation

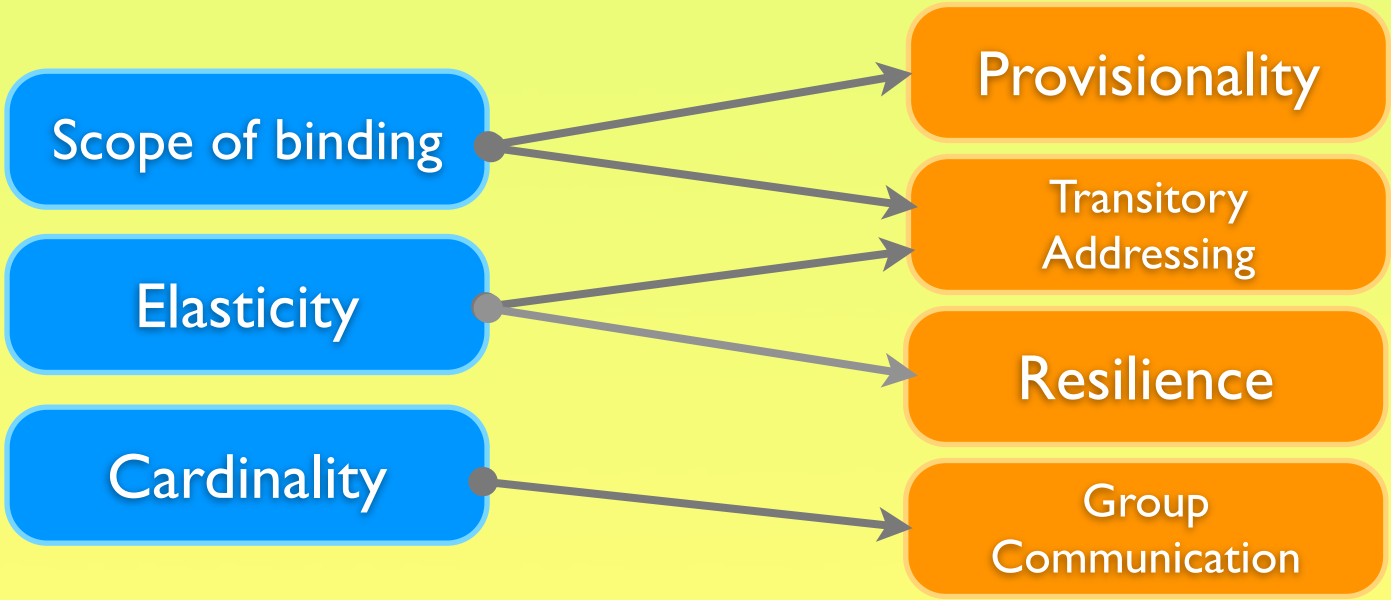
- Local **proxy** for remote service
- Performs discovery on behalf of its client



AMBIENTTALK

Conclusion

- Pervasive computing requires novel language abstractions!
- Ambient references: **remote object references** for mobile networks


Scope of binding
Elasticity
Cardinality
Provisionality
Transitory Addressing
Resilience
Group Communication