Waterken Web

Secure Interoperation Using a Capability Messaging Protocol

Tyler Close, Founder
Waterken Inc.
http://www.waterken.com/

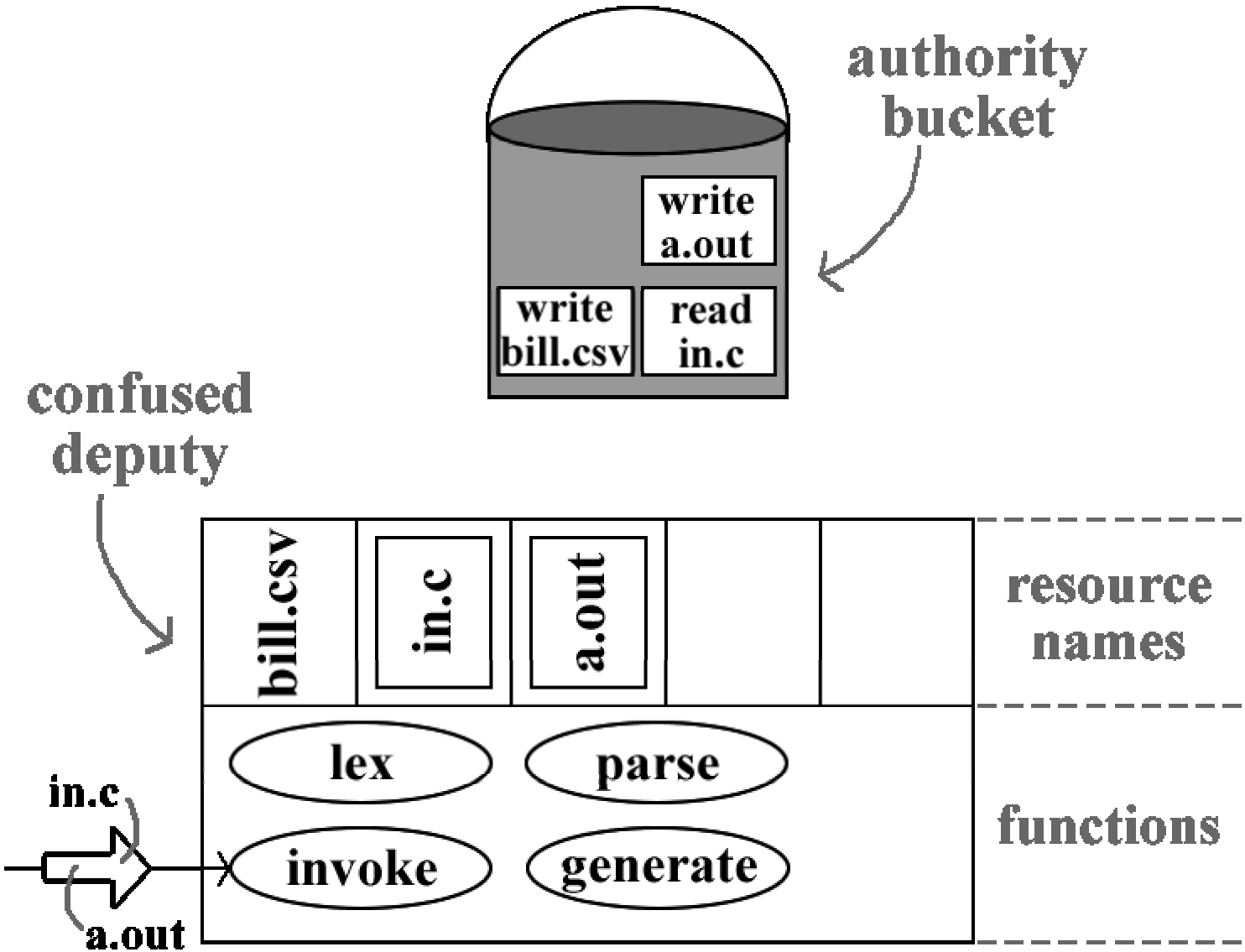
Overview

- The problem domain
- Capability-based security
- What is achievable?
- The web-calculus
- The web-amp
- Doc Model + Schema
- HTTP binding
- RDB Webizer

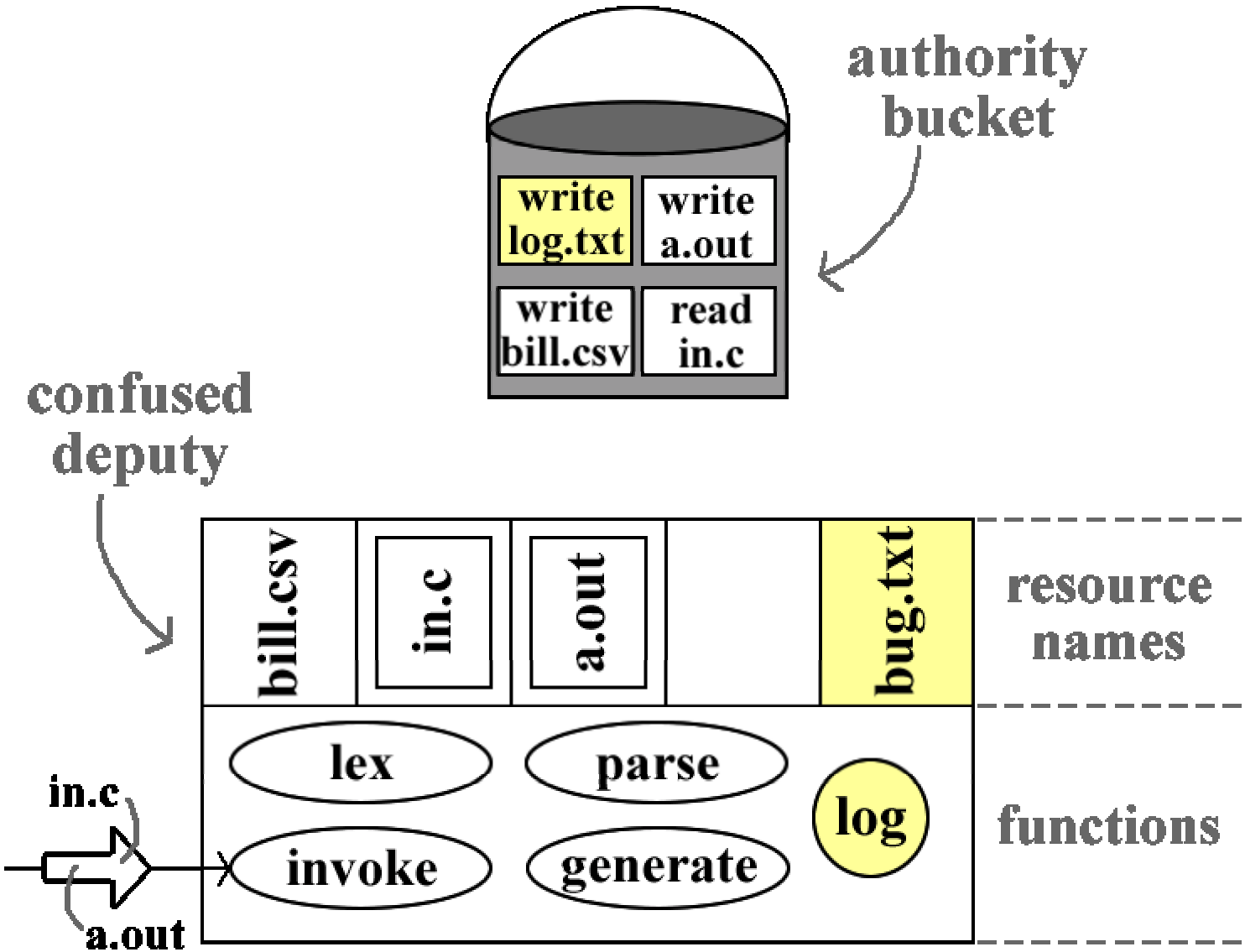
Problem Domain

- “Open Office” –like communication between mutually suspicious parties
- Confused deputy attack
- A web service is a deputy

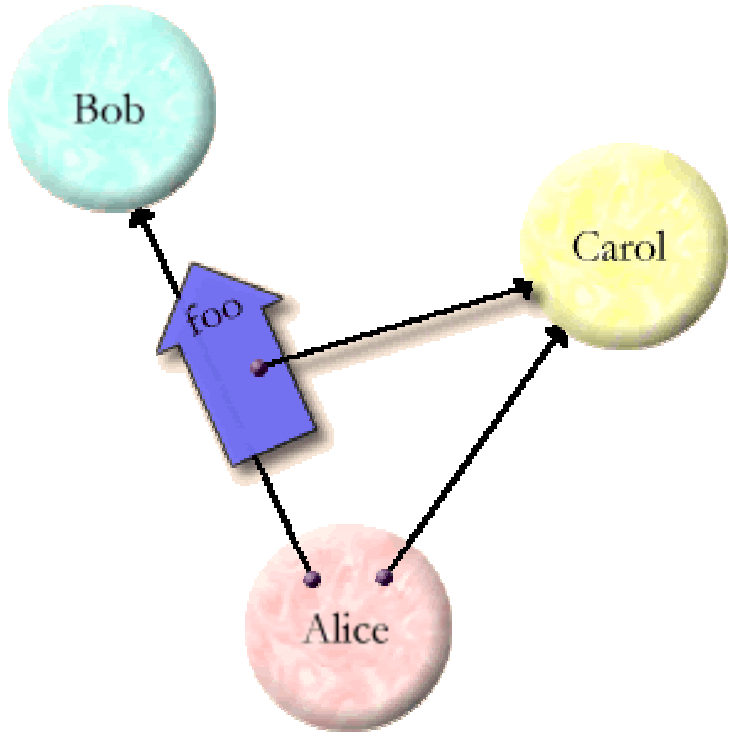
Confused Deputy



Confused Deputy 2

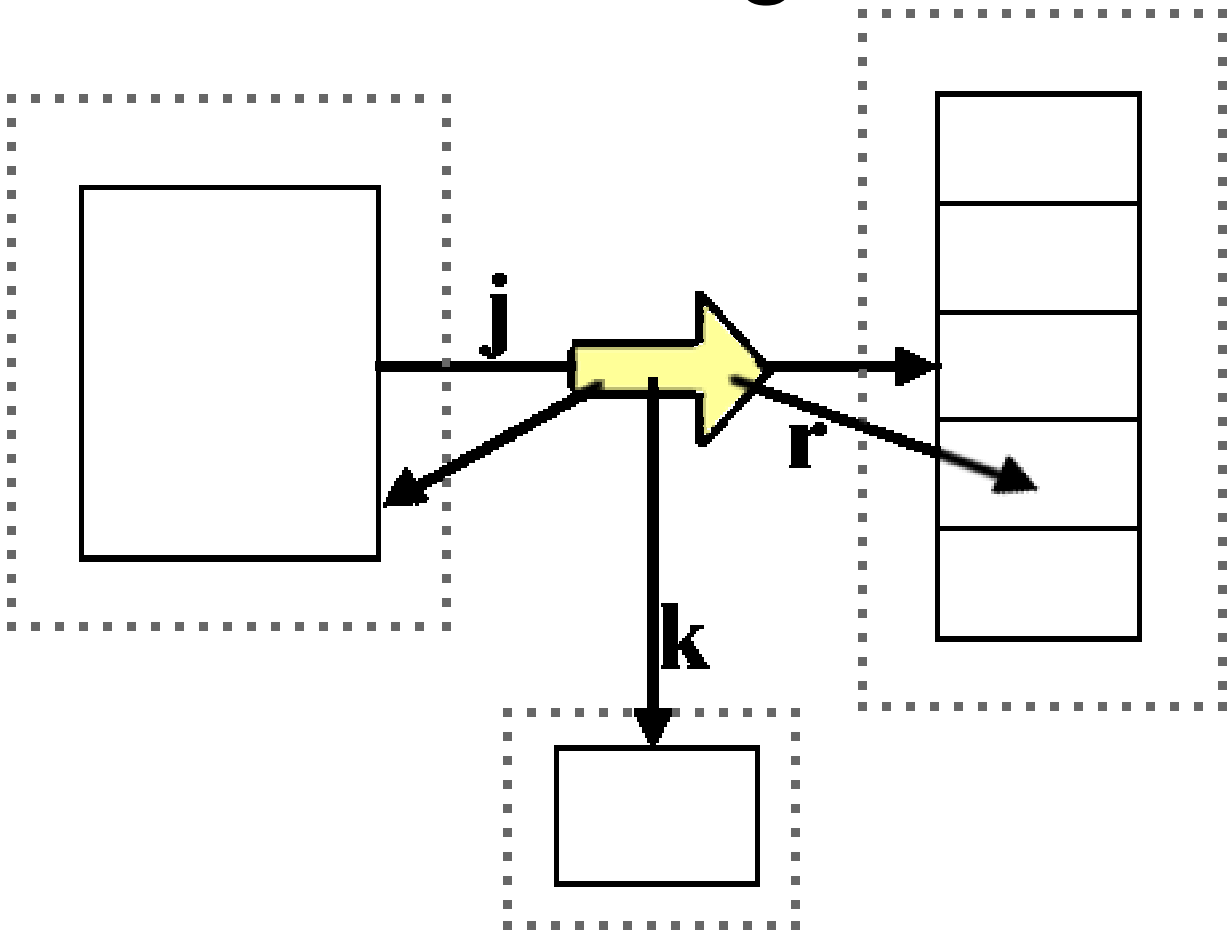


Capability Security



- The origin
- The rules
- The requirements

The Origin



Dennis, Van Horn. Section VII: Protected Entry Points. Programming Semantics for Multiprogrammed Computations, 1965.

The Rules

1. Only connectivity begets connectivity
2. Absolute encapsulation
3. There is nothing other than #1 and #2

See: “An Ode to the Granovetter Diagram”

http://www.erights.org/elib/capability/ode/

The Requirements

- Encapsulation
- Unforgeable references
- Target authentication
- Private communication

Leverage

- **Distributed security before local security**
  - Enabled by Principle of Least Privilege
- **Programming model independence**
  - Security model is one with reference model
- **No centrally administered namespace**
  - Reference namespace = authority namespace

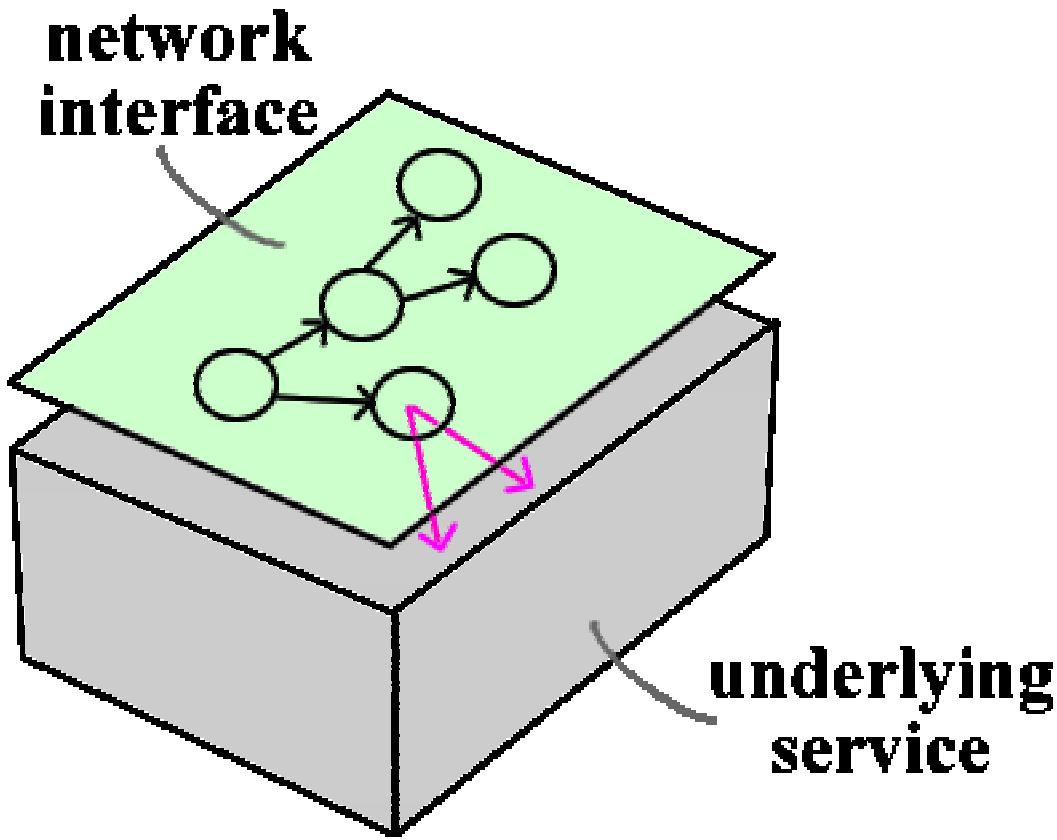
Web-Calculus

- Design goals
- Structure
- Operation
- Implementation

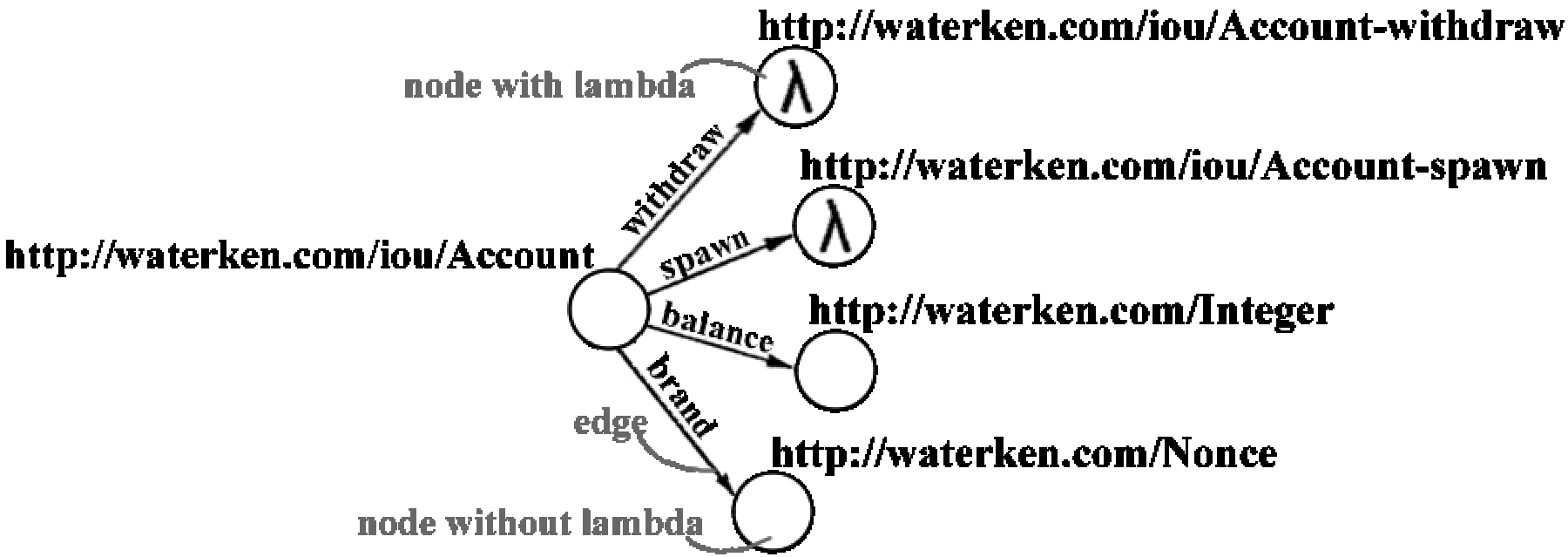
Design Goals

1. Capability-based access control
2. Meta-model
3. Partial understanding
4. Interface refactoring

Layered Model



Structure



GET **POST** **EXPECT**

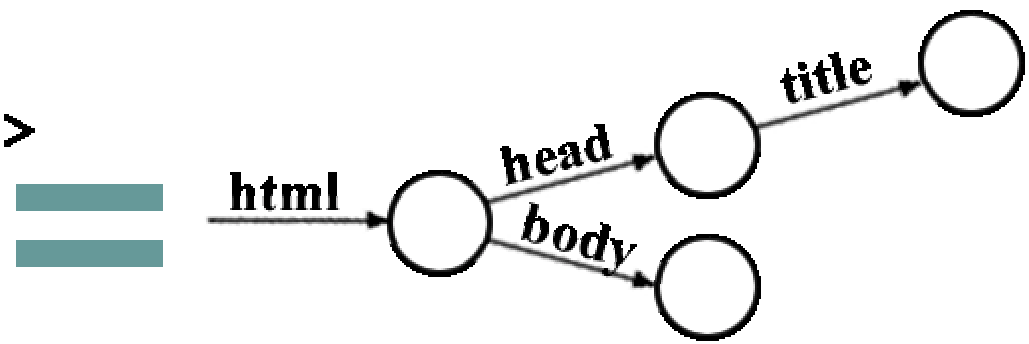
EXTRACT **SETTLE**

Implementation

- Hypertext
- REST
- RPC
- DEM
- NOM
- CCP
- UML

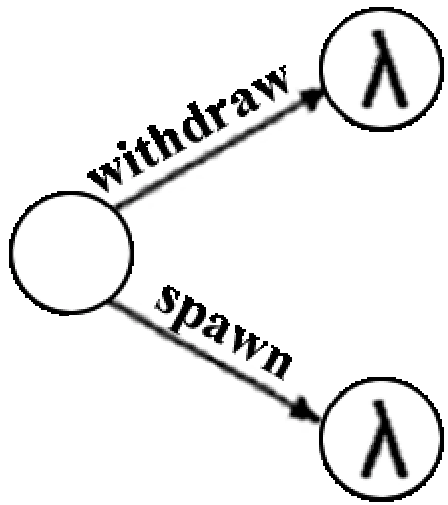
Hypertext

```
<html>
 <head>
  <title>Welcome</title>
 </head>
 <body>
  My home page.
 </body>
</html>
```

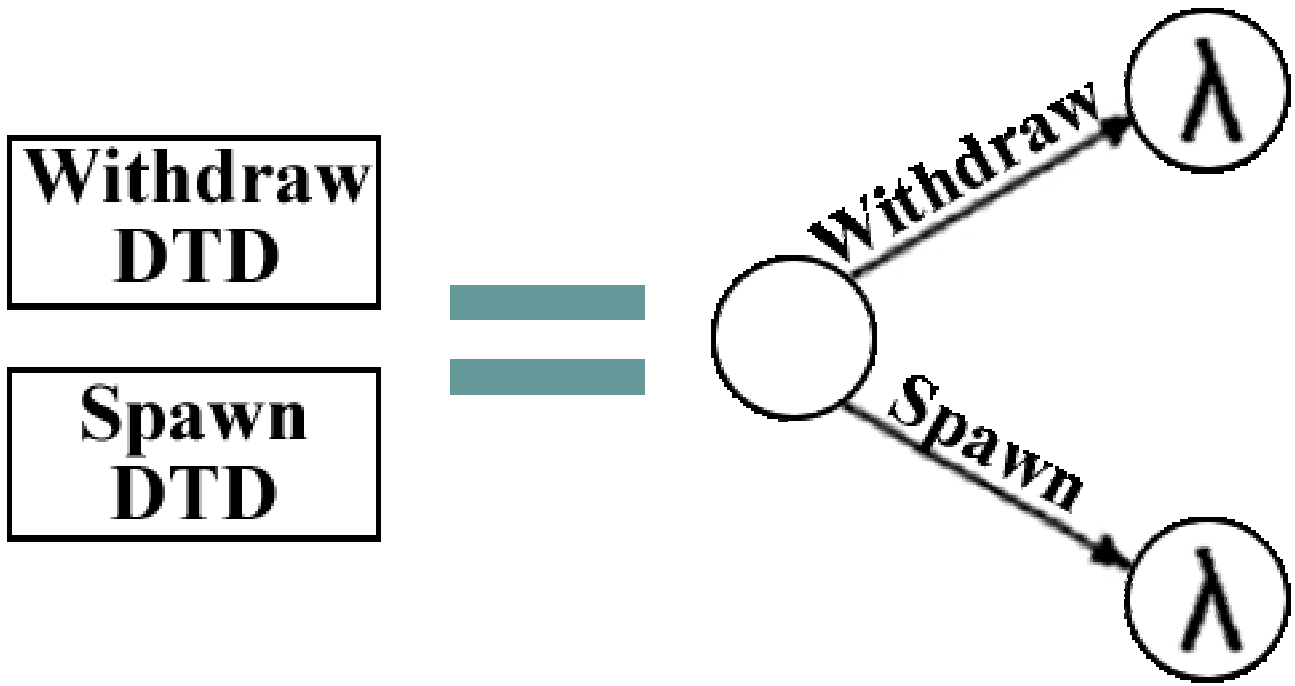


RPC

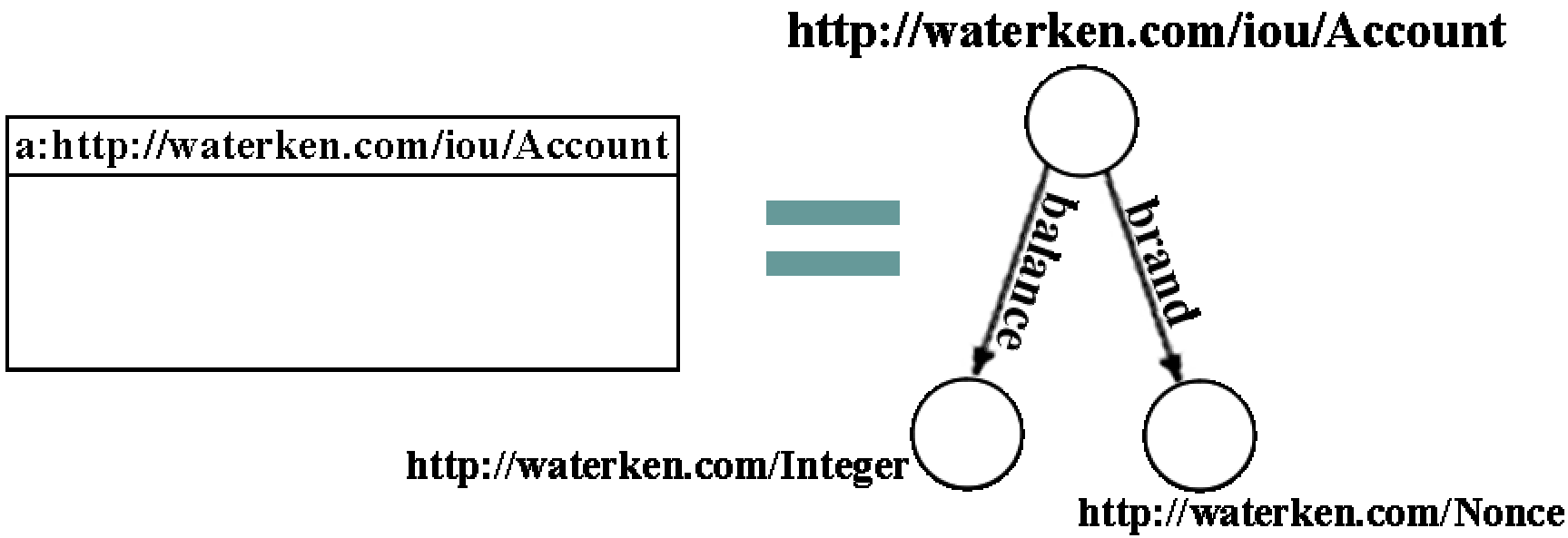
```
withdraw(account,integer);
spawn(account);
```



DEM



NOM



UML

Web-Amp

- Design Goals
- Capability URI
- Envelope
- Message pipelining
- Reference passing

Design Goals

1. The capability semantics of edges is preserved.
2. A secure model for handling transmission failures is supported.
3. Features that magnify a denial-of-service attack are not required.
4. The protocol is easy to understand and use.
5. A simple implementation of the protocol is possible.
6. Interoperation in a heterogenous network environment is supported.

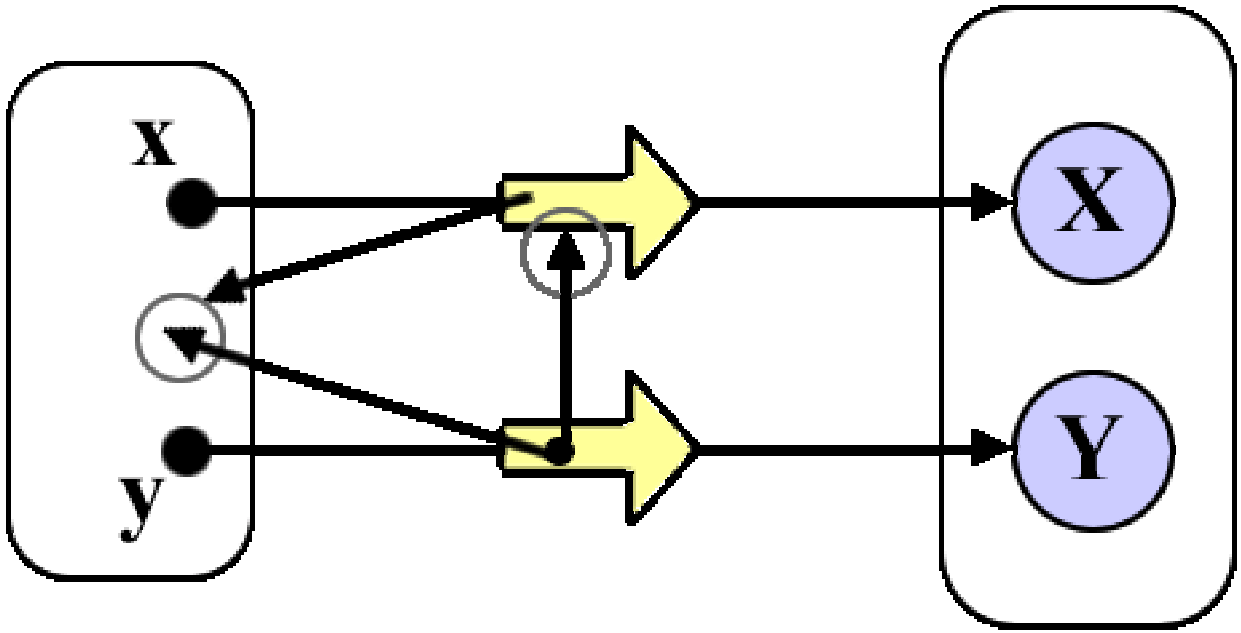
Capability URI

- An unguessable GUID identifying the remote node
- A means to identify the host on which the remote node currently resides.
- A means to authenticate the host
- A means to establish a secret message transport
- A “soft reference”

Envelope

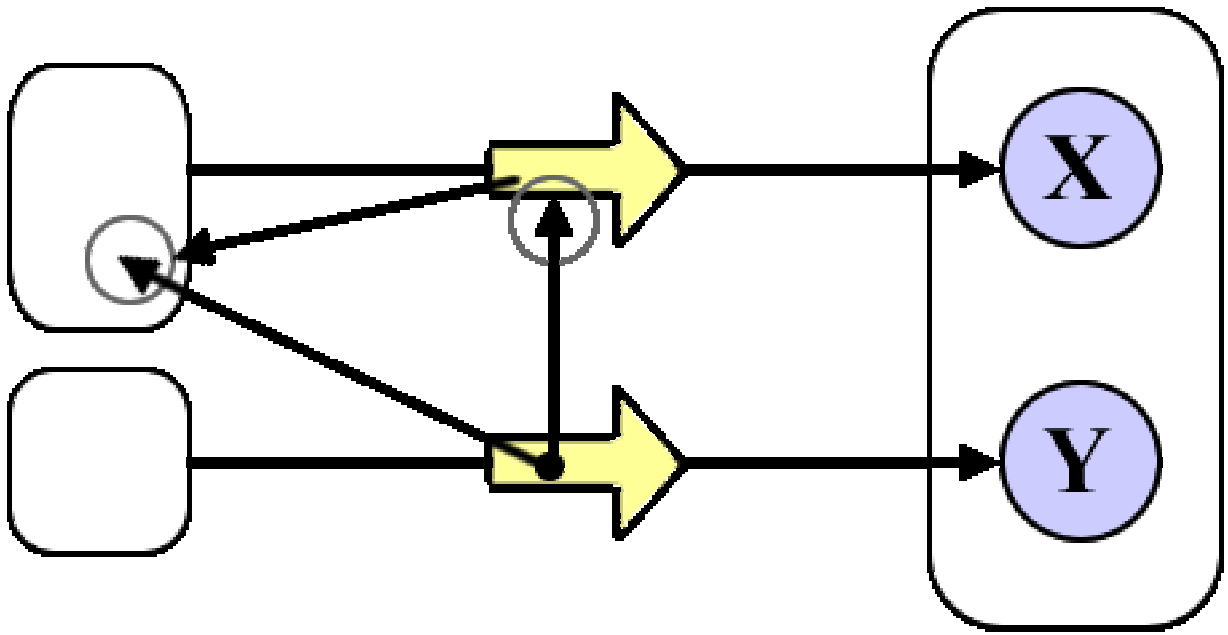
- source node — capability URI
- target path — ‘**`foo/bar/`**’
- operation — **`POST`**
- argument list — **`(5, “hello”)`**
- return resolver — capability URI

Message Pipelining




```
promise_guid = to_base32(md5(to_ascii(source_guid +
                                      resolver_guid)))
```

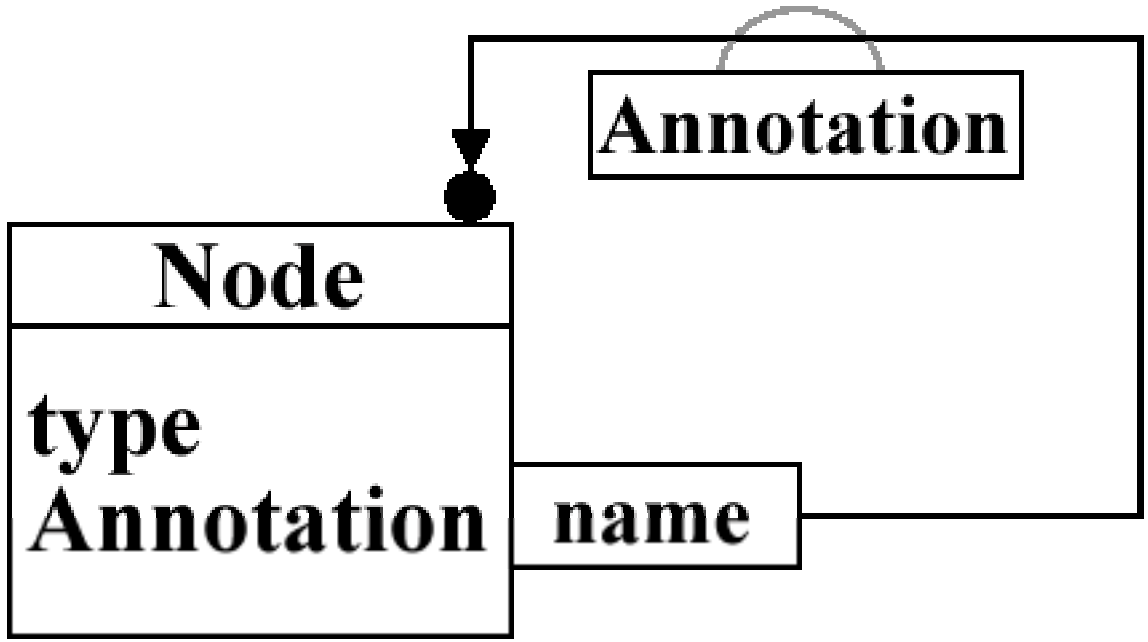
Reference Passing



Waterken Doc

- Model
- Schema
- Pointer
- XML
- code
- www-form

Doc Model



Doc Pointer

- http://waterken.com/doc/pointer/Link
- http://waterken.com/doc/pointer/Embed
- http://waterken.com/doc/pointer/ID
- http://waterken.com/doc/pointer/Broken
- http://waterken.com/doc/pointer/Pipeline

Doc XML

```
<doc schema=http://waterken.com/time/local/Clock>
        <super schema=http://waterken.com/time/Clock>
                <now>
                        <date>
                                <day>27</day>
                                <month>2</month>
                                <year>2003</year>
                        </date>
                        <time>
                                <hour>13</hour>
                                <minute>51</minute>
                                <second>36</second>
                        </time>
                        <zone>
                                <hours>5</hours>
                                <minutes>0</minutes>
                        </zone>
                </now>
        </super>
</doc>
```

HTTP Binding

Capability URL:

https://www.waterken.com/example/2lk3lsjlgk09djxkqew92ldi10s

HTTP Request:

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

Operation Envelope:

Envelope = Operation SP
Source-URI?path=Path&mid=Resolver-GUID SP
HTTP-Version CRLF

RDB Webizer