Introduction to Eve

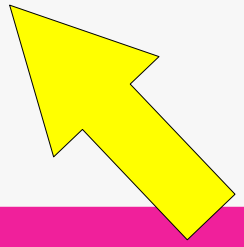
a multipurpose, web based agent platform

Almende B.V.
2012-06-08

http://almende.github.com/eve
http://www.almende.com

Agent platforms

- AgentScape http://www.agentscape.org
- Goal http://mmi.tudelft.nl/trac/goal
- Jade http://jade.tilab.com
- Janus http://www.janus-project.org
- Jason http://jason.sourceforge.net
- MadKit http://www.madkit.org
- Our own platforms CAL and Emerge
- Eve http://almende.github.com/eve



Overview

- Concept
- Modules
- Demo
- Hello World
- Open ends

Design considerations

- Keep it simple
- Scalable, decentral, web based
- Use existing infrastructure and protocols
- Create an open platform, that works across system boundaries
- Do not enforce a single programming language or deployment environment
- Mimic human society and interaction
- Inspired by RESTful JSON API's

Concept

- The “Eve World” consists of the World Wide Web
- Each Agent has its own unique URL
- Agents communicate using JSON-RPC over HTTP POST requests

Protocol: JSON-RPC

Url	HTTP POST http://myserver.com/agents/type/id

Request	{
	"id": 1,
	"method": "add",
	"params": {
	"a": 2.2,
	"b": 4.5
	}
	}

Response	{
	"id": 1,
	"result": 6.7,
	"error": null
	}

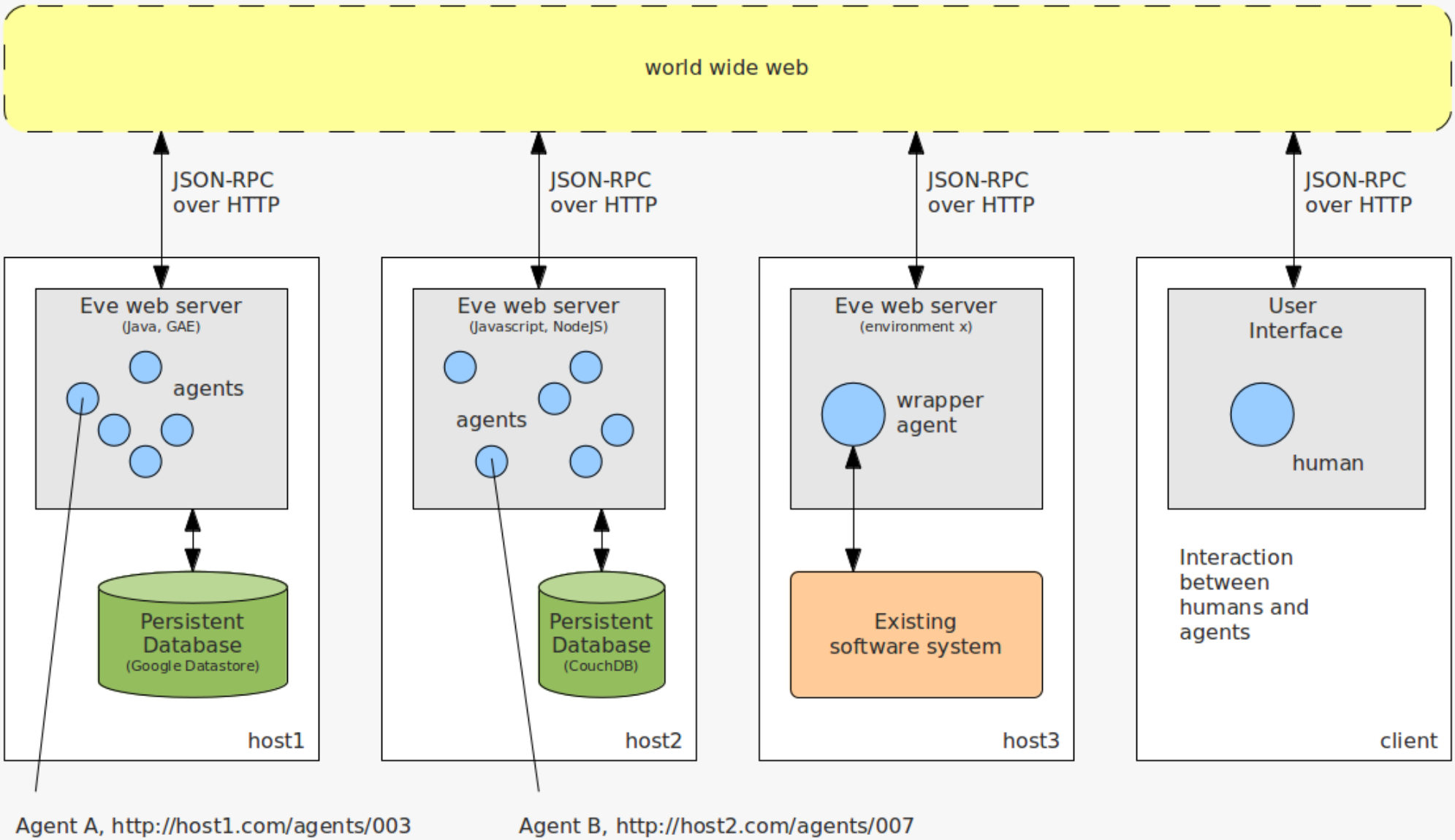
Resources

Common resources need to be standardized

- Calendar Event
- Geo location
- Address
- Activity
- ...

Eve

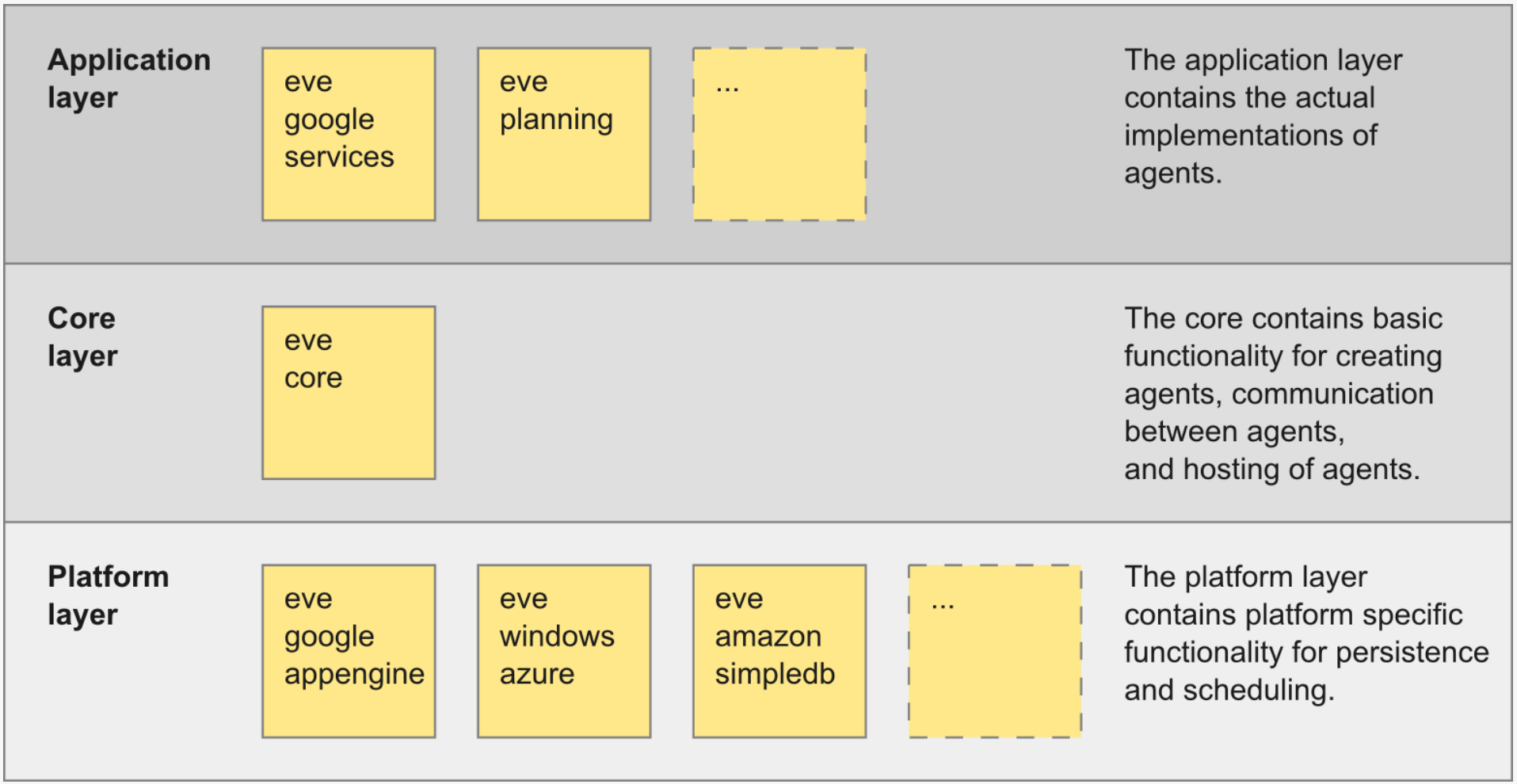
a web-based agent platform



Implementations

- The protocol is so simple – you don't need a library and can write your own interface in no time
- Java implementation
- Node.js implementation
- More to come...

Java modules



Eve Core

- Servlets: host a single or multiple agents
- Resources: activity, geolocation, etc.
- Agents:
  - Stateless
  - Context
  - Event driven
  - Subscribe to events
  - Scheduler

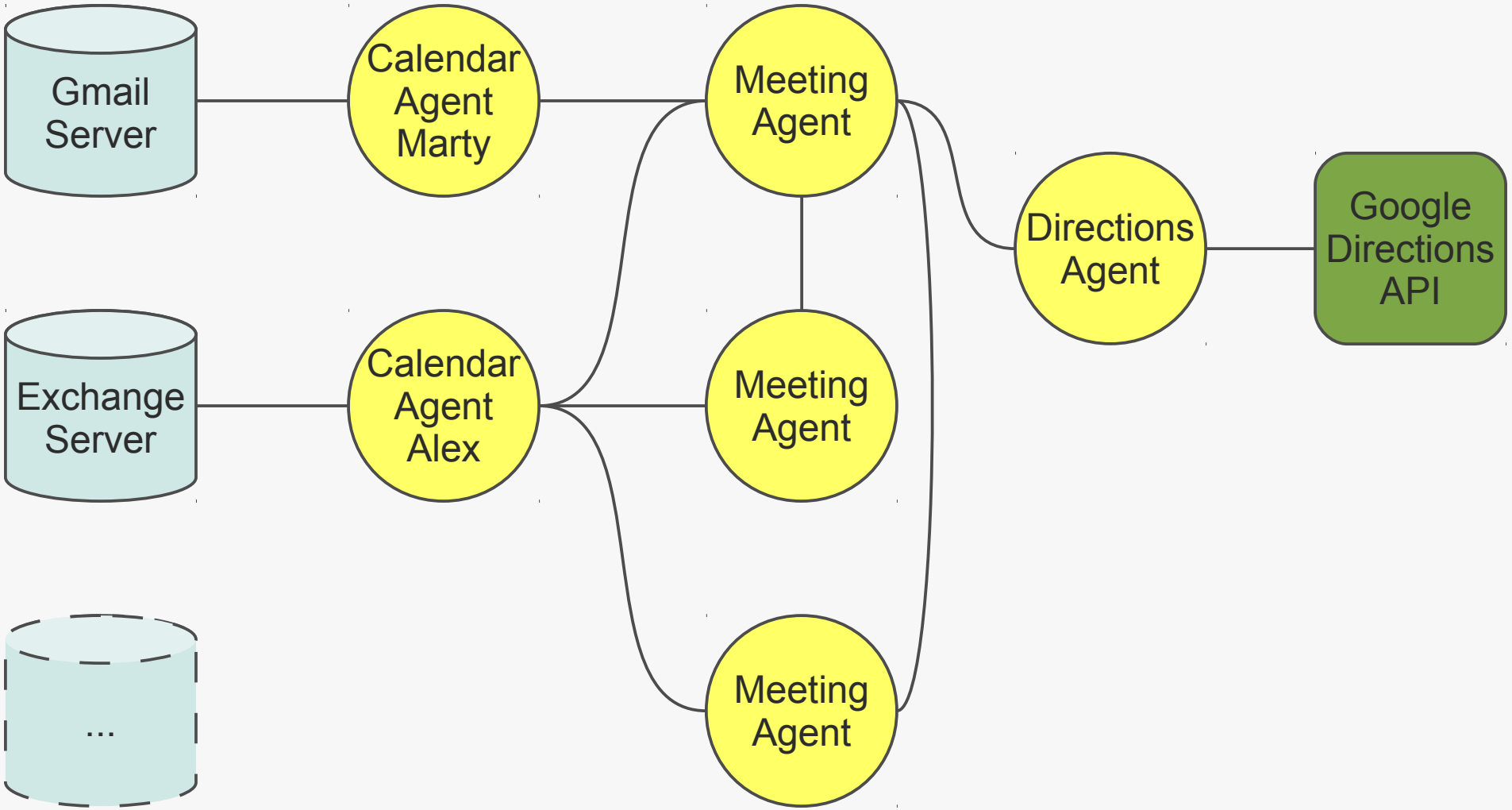
Eve Planning

- First implementation with real agents
- Acts on the calendaring/planning domain
- Implements:
  - CalendarAgents for accessing various calendaring systems
  - MeetingAgents for dynamically planning meetings
  - DirectionsAgents for calculating travel time
- Is being implemented in Ask's Paige

MeetingAgent Activity

- Description
- Constraints
  - People/resources
  - Location
  - Time
- Status
  - Current status: planned, executing, canceled
  - Planned status: resources, location, time

Example scenario



Demo time

- Peer-to-peer chat agents

  http://eveagents.appspot.com/agents/chatagent/1/
  http://eveagents.appspot.com/agents/chatagent/2/
  http://eveagents.appspot.com/agents/chatagent/3/

- CalendarAgent and MeetingAgent

  http://eveagents.appspot.com/demo/

Getting Started

- Create a Java (web) project
- Download the Eve libraries, add them to the build path of your project
- Add an Eve config file, configure a servlet
- Create your own agents

HelloWorldAgent (1/2)

```
public class HelloWorldAgent extends Agent {
    public String welcome(@Name("name") String name) {
        return "Hello " + name + "!";
    }

    @Override
    public String getVersion() {
        return "0.1";
    }

    @Override
    public String getDescription() {
        return "This agent can do this and that for you.";
    }
}
```

HelloWorldAgent (2/2)

```
public class HelloWorldAgent extends Agent {
    // ...

    public void setUsername(@Name("username") String username) {
        getContext().put("username", username);
    }

    public String getUsename() {
        return getContext().get("username", String.class);
    }

    public void getOthersUsername() {
        String url = "http://server/agents/helloworldagent/othersid";
        String method = "getUsername";
        ObjectNode params = null;
        String username = send(url, method, params, String.class);
        System.out.println("Others username is " + username);
    }
}
```

Check it out

http://almende.github.com/eve/

- Libraries
- Documentation
- Getting started
- Examples

Current open ends

- How to find agents? Centralistic directory services vs. mimicing the way humans learn to know each other
- Authorization: traditional models vs. trust/reputation models
- Handling (binary) resources: make a hybrid solution with RESTful API?
- Algorithms for dynamically planning activities
- Algorithms for negotiation between agents

What makes Eve unique?

- Simple
- Open environment
- Scalable, decentral
- Utilizes modern web & cloud solutions
- Defines the protocol, does not dictate how to built or deploy your application

Discussion time

beer please...