Towards Context-Aware Adaptable Web Services

Markus Keidl

Universität Passau
Fakultät für Mathematik und Informatik
D-94030 Passau
`keidl@db.fmi.uni-passau.de`

Alfons Kemper

TU München
Fakultät für Informatik
D-85748 Garching
`alfons.kemper@in.tum.de`

Outline

- Motivation
- ServiceGlobe
- Context Framework
    - Infrastructure: Context Model and Life-Cycle
    - Context Processing
    - Context Processing Instructions
- Summary

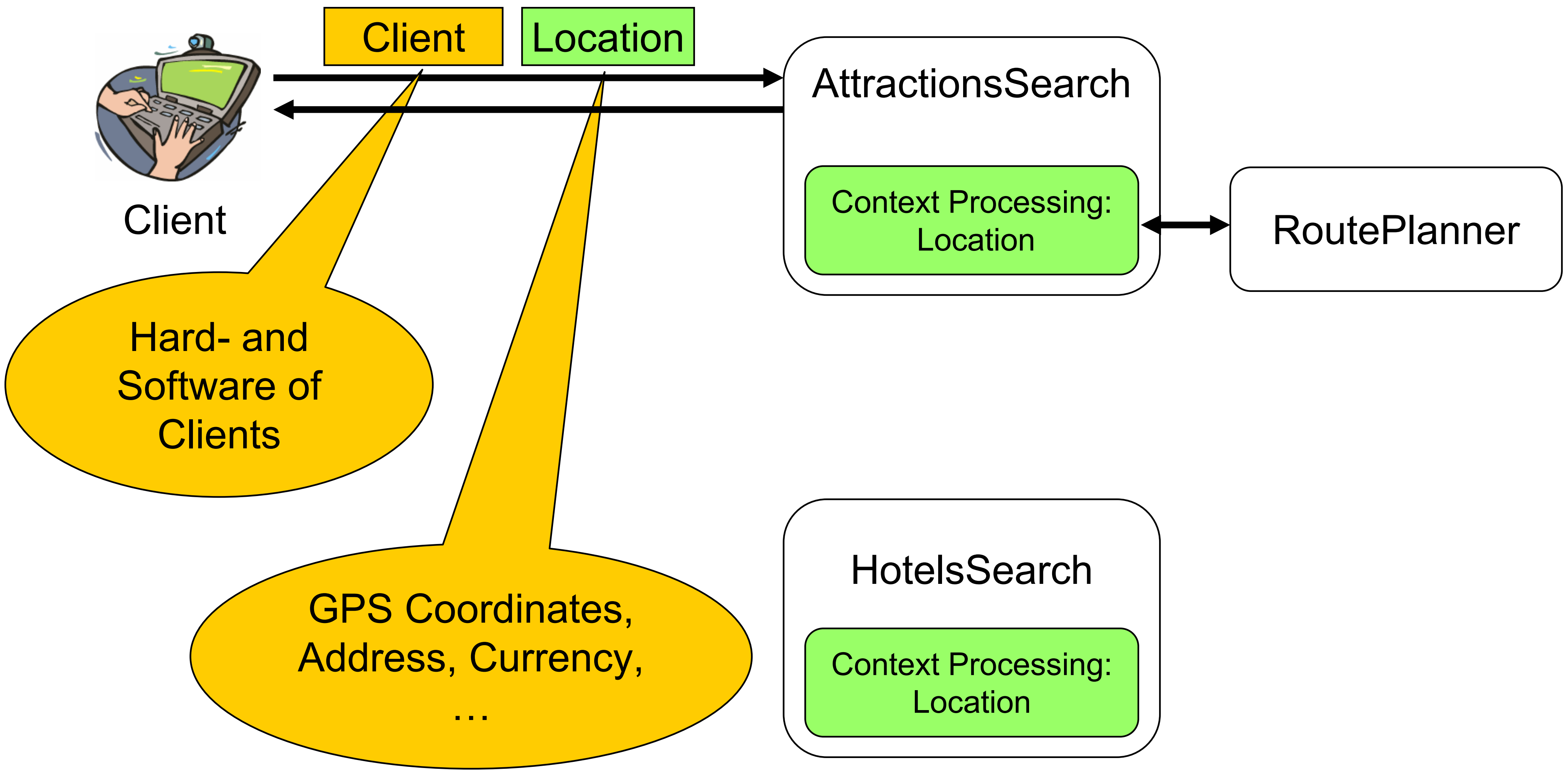
Motivation

- Web services on the Internet:
  - Large number of clients
  - Heterogeneity of client capabilities and number of methods for accessing Web services
- Pervasive Computing:
  - Increasing number of ubiquitous, connected devices
- Goal:
  Framework that facilitates the development and deployment of Web services aware of this

The Context Framework

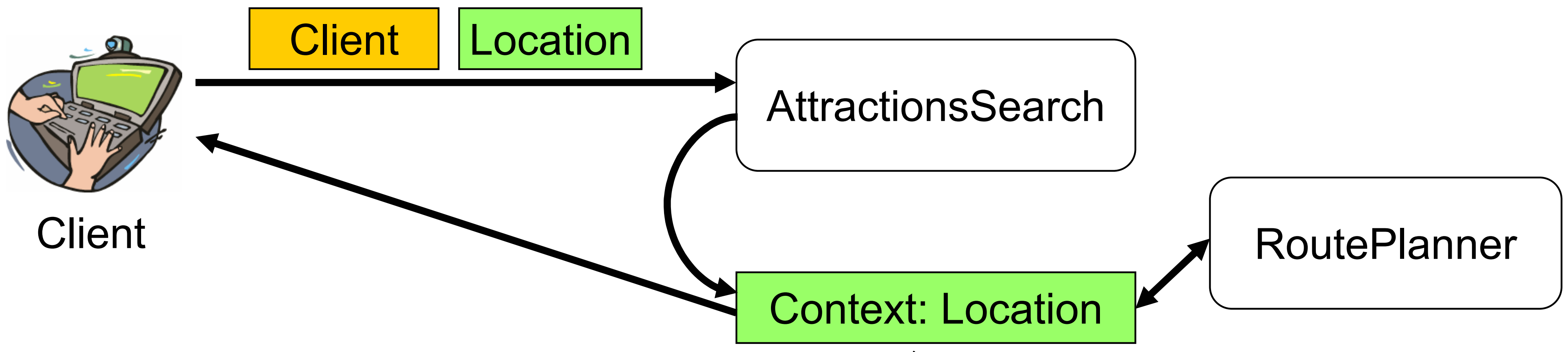
- Context: Information about clients and their environment that is used by Web services to provide clients with a customized and personalized behavior
- Examples:
  - Location of a client: GPS coordinates, address, time and time zone, currency, …
  - Device type: Hard- and software of client

Motivating Scenario



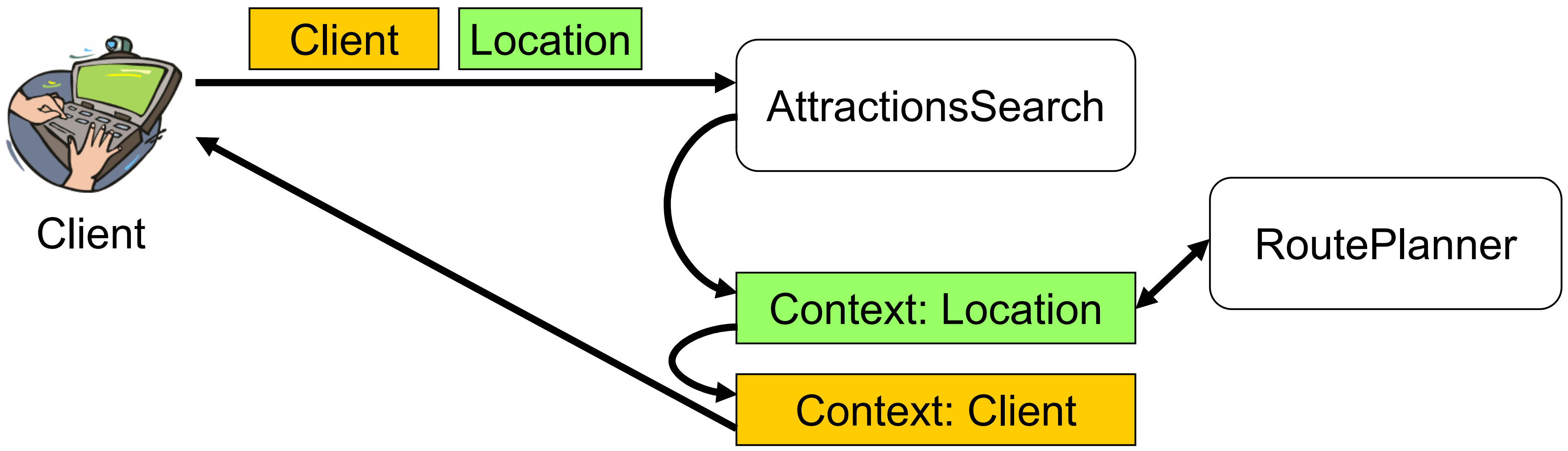
Client
Location
Client
AttractionsSearch
Context Processing: Location
RoutePlanner
Hard- and Software of Clients
GPS Coordinates, Address, Currency, …
HotelsSearch
Context Processing: Location

Motivating Scenario



Requirements:
- Isolation of Functional Duties
- Transparency
- Automatic Processing
- Generic Solution

Motivating Scenario



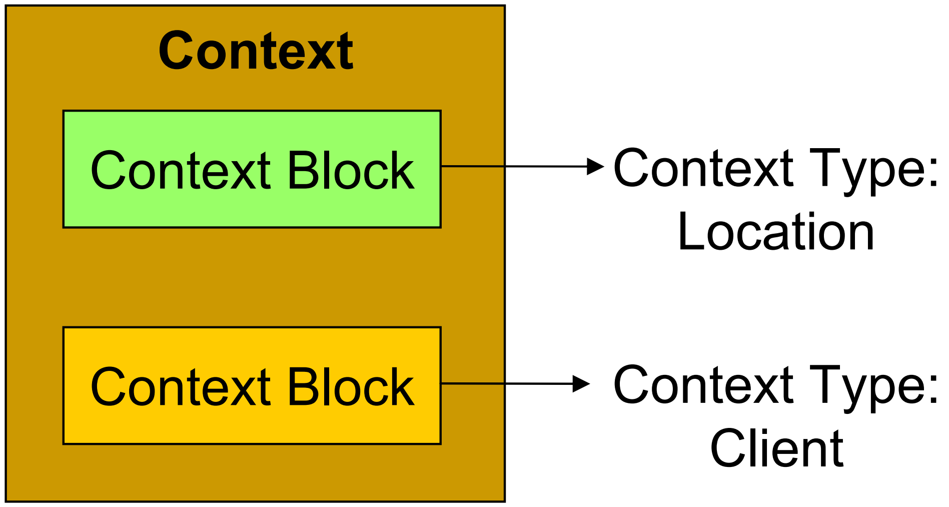
Features of the Context Framework

- Separation of functional duties into external components: *context plugins and context services*
- Transparent and automatic usage of these components
- Generic solution, i.e., components are usable for a variety of Web services

Web Service Platform ServiceGlobe

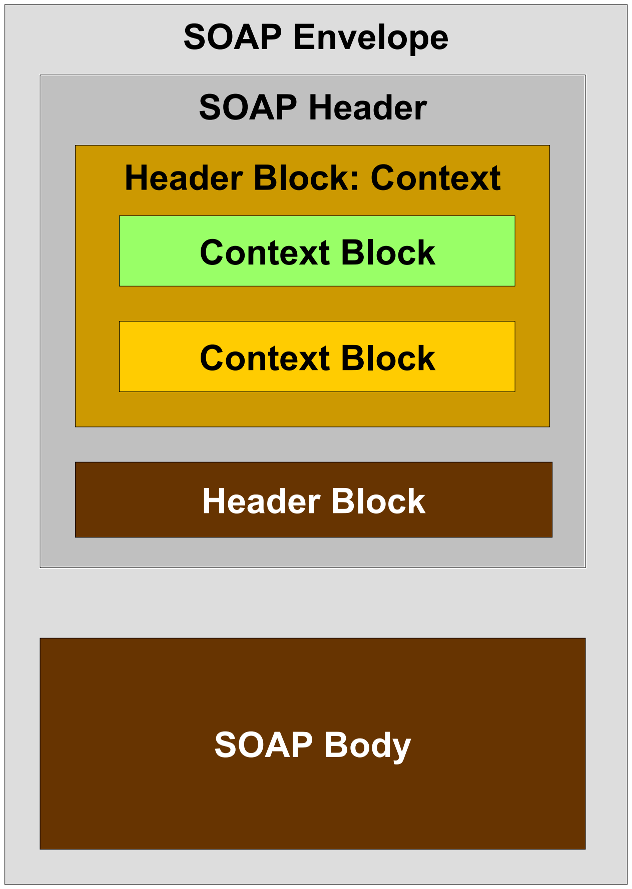
- Research platform for mobile Web services
  - Web services are mobile code
  - Fully implemented in Java
  - Based on standards like XML, SOAP, UDDI, WSDL,…
- Standard functionality of a service platform
  - Transaction system
  - Security system

Context Model



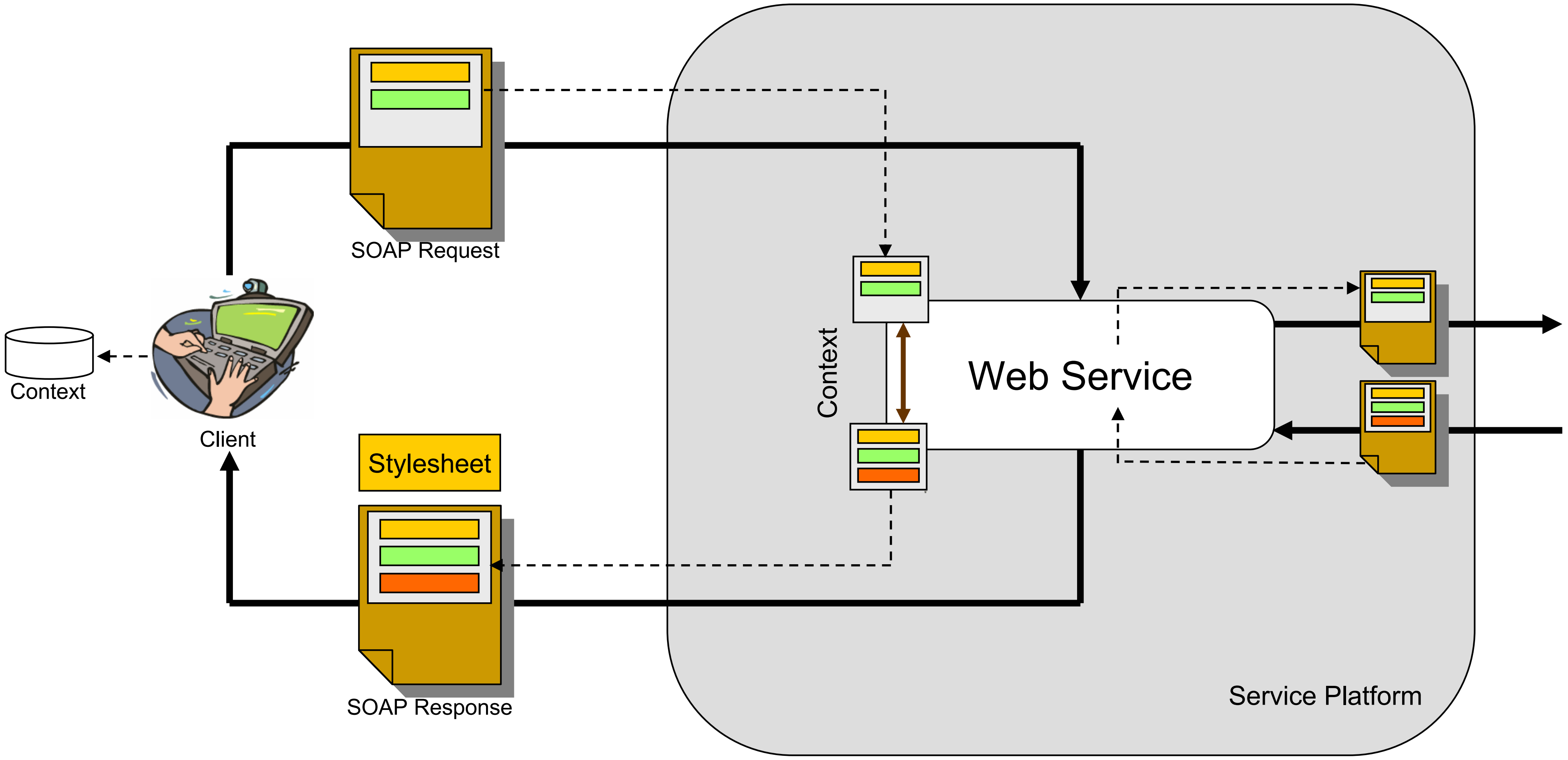
- Context consists of several context blocks
- A context block is associated with one context type
- A context type defines the type of context information in a context block, e.g., location, client device
- At most one context block is allowed for a context type within a context
- Context is transmitted as a SOAP header block

Context in a SOAP Message

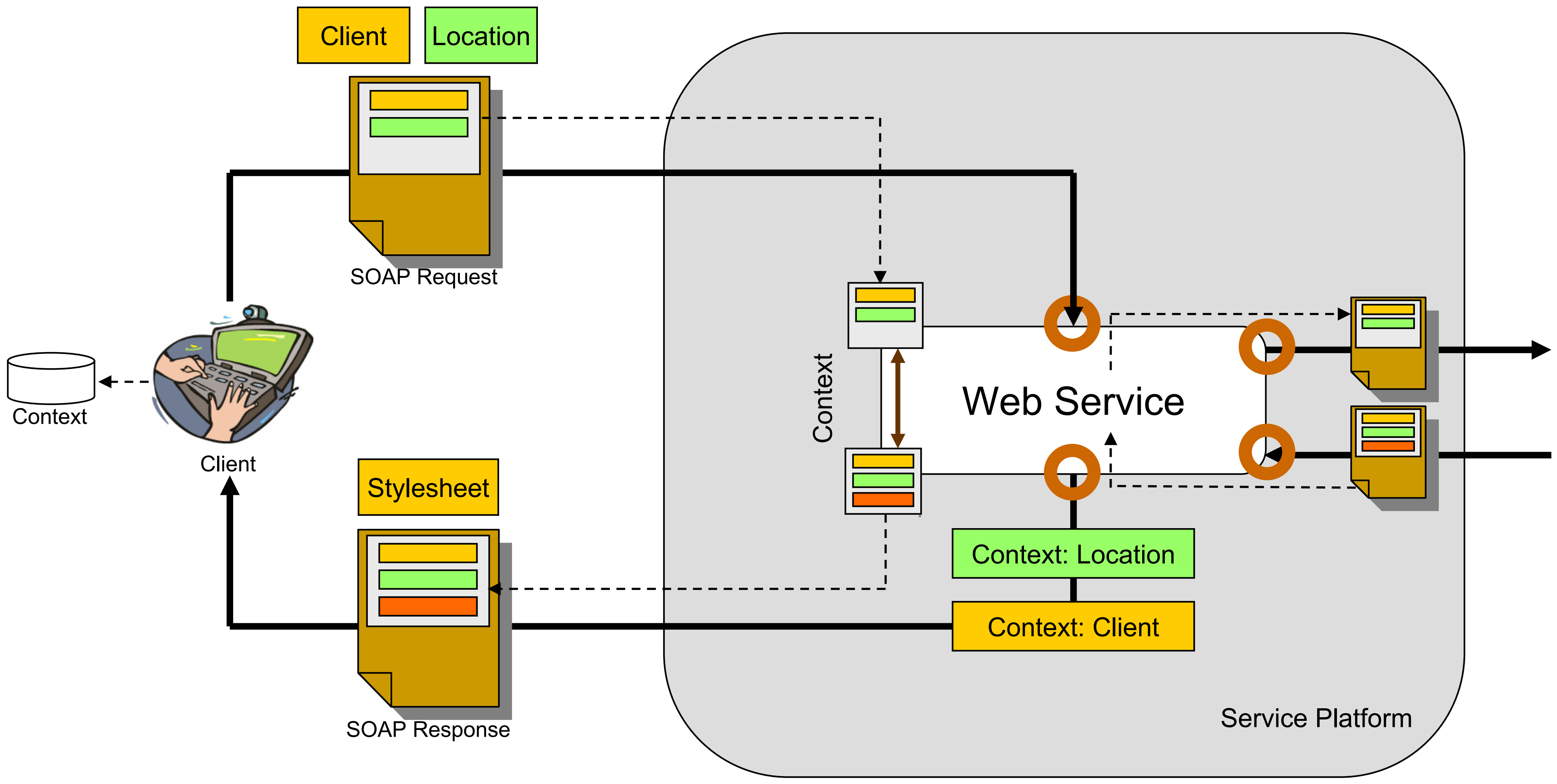


```
<env:Envelope xmlns:env="...">
  <env:Header>
    <Context
       xmlns="http://sg.fmi.uni-passau.de/context">
      <Client>
        <Hardware>
          <Defaults>
            http://example.com/PDA
          </Defaults>
          <ScreenSize>320x320</ScreenSize>
          <IsColorCapable>Yes</IsColorCapable>
        </Hardware>
      </Client>
    </Context>
  </env:Header>
  <env:Body>
    <!-- serialized object data -->
  </env:Body>
</env:Envelope>
```

Context Life-Cycle



Context Life-Cycle



Context Processing

- Explicit Context Processing
  - Context is accessed explicitly using the Context API
  - Components: Web services and clients
- Automatic Context Processing
  - Pre- and post-process SOAP messages automatically based on their context
  - Components: Context plugins and context services

Automatic Context Processing

- Context operations:
  - Pre-process a Web service request
  - Post-process a Web service response
  - Post-process an outgoing Web service message (request to another Web service)
  - Pre-process an incoming Web service message (response of another Web service)
- Processing of context blocks within a SOAP message in arbitrary order
- Invocation of suitable components: context plugins and context services

Components for Automatic Context Processing

- Context plugins and context services
  - Associated with one dedicated context type
  - Input: context block and message
  - Output: (possibly modified) message
- Advantages:
  - Generic solution
  - Extensible solution
- Restriction:
  - No influencing of internal control flow of Web services

Context Plugins and Context Services

- Context plugins: Java objects implementing the **`ContextPlugin`** interface
  - Installation on local host
  - Access to internal data structures of service platform
- Context services: Web services implementing the **`ContextService`** WSDL interface
  - Available anywhere on the Internet
  - Extendible at runtime
  - Only access to context and Web service messages

Context Processing Instructions

- Problems:
  - Which components should process a context block?
  - On which hosts should a context block be processed?
- Solution: Specification of
  - Instructions for context plugins and context services
  - Processing guidelines

  for every context type individually

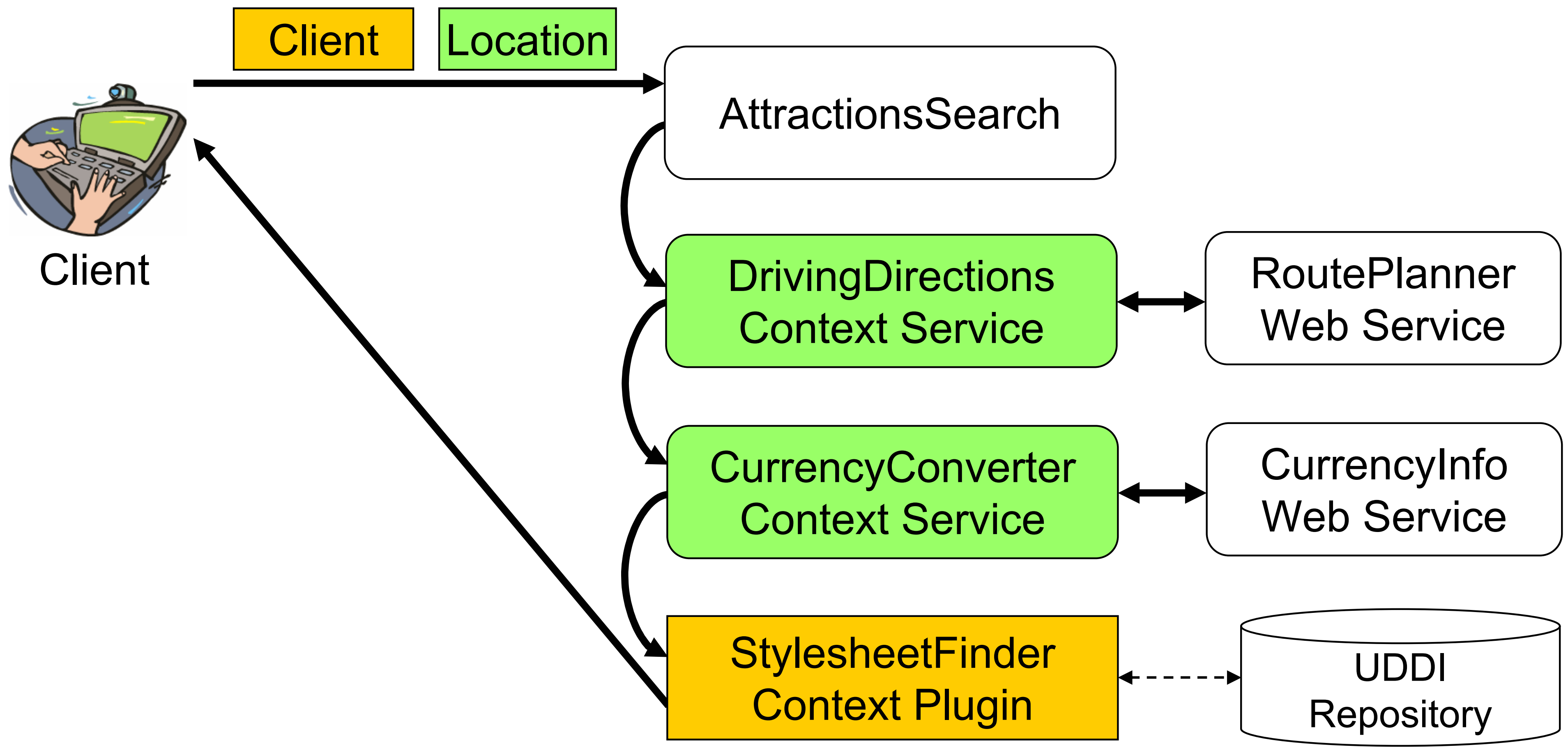
Example of Context Processing Instructions

```
<ContextProcessingInstructions>
  <ContextType ID="http://sg.fmi.uni-passau.de/context:Location">
    <ContextService>
      <AccessPoint useType="http">
        http://example.org/services/CurrencyConverter
      </AccessPoint>
      <ContextOperations>post</ContextOperations>
    </ContextService>
    <ProcessingGuideline>
      <ServiceHost>Next</ServiceHost>
      <ComponentTypes>
        ContextService
      </ComponentTypes>
    </ProcessingGuideline>
  </ContextType>
</ContextProcessingInstructions>
```

Providing Context Processing Instructions

- Within a SOAP message’s context:
  - Insertion of context processing instructions as self-contained context block
- Within a Web service‘s UDDI metadata
- Within UDDI:
  - Association of context services to the tModel of their context type
  - Context framework searches for suitable context services for a specific context type

Motivating Scenario



Summary

- Context Framework
  - Context Infrastructure: Model and Life-Cycle
  - Context Processing
    - Explicit Context Processing
    - Automatic Context Processing:
      Context plugins and context services
  - Context Processing Instructions
    - Allow to control the context processing