

Web Mashups

OpenAjax Hub

Rainer Villido

Rainer Villido

- Magistrant Tartu Ülikoolis
- STACC – Tarkvara Teadus ja Arenduskeskus
 - Rakenduslik teadusprojekt Regio ja Delfiga

Web 2.0

- Kasutajakesksus
- Kasutajad loovad sisu
- Informatsiooni jagamine
- Süsteemidevaheline koostöö
- Kõikvõimalike komponentide koostoimimine
- The Web as Platform

Widgets (vidinad)

- Iseisvad väikesed veebirakendused
- Taaskasutatavad
- Lihtsad kasutada (tavakasutajale)
- Jooksevad erinevates keskkondades
 - Veeb
 - Desktop
 - Mobiil

Vidinate demo

Widgets (vidinad)

- Erinevad platformid
 - Su enda koduleht
 - iGoogle & Google Desktop
 - Apple Dashboard
 - Windows Sidebar
 - Facebook
 - Yahoo! Widgets
- (Erinevate platformide tehnoloogiad pole veel eriti ühilduvad – tehnoloogia ja standardid on alles kujunemas)

Widgets (vidinad)

- Platformide vähene ühilduvus
 - Erinevad pakendamismaadid
 - Erinevad API-d
 - Erinevad turvamudelid
 - veebirakendus ei pääse arvuti ressurssidele ligi, kuid desktopi või mobiili vidin võib pääseda
 - Näiteks info CPU, RAM, HDD kohta
 - Google Widgets, Opera Widgets, Windows Gadgets, Dashboard Widgets (Apple), Yahoo! Widgets, Netvibes UWA, OpenAjax Widgets

Widgets (vidinad)

- Ühiste standardite loomine
- W3C Widgets 1.0 standard
- Bondi OMTP standard mobiilidele
- OpenSocial sotsiaalvõrgustikele

Widgets (vidinad)

- Kõik vidinad on olemuselt veebirakendused
- HTML leht
- CSS – kujundus
- JavaScript – rakenduse loogika
- Soovikorral peab JavaScripti abil enda rakendusserveriga ühendust (AJAX)

Mashups

- Kasutajad loovad vidinad kokku ühele lehele
- Olemasolevate ressursside kombineerimine
- Lõppkasutajad loovad endale ise rakendusi
- Individuaalsed (loogika ja kujundus)
- Arendajad loovad vidinaid
- Uus arendusmudel
- Kiire

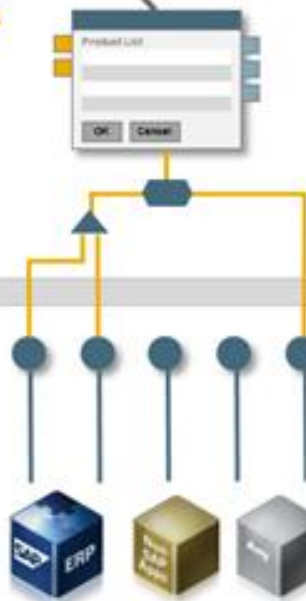
Mashup

- **Composition of Widgets**
(Wiring) to enterprise-class applications



Gadget

- **User-friendly Building Blocks**
- Provide graphical and simple user interaction mechanism
- **Abstracting** from the complexity of the underlying resources



Resource

- **Core Building Blocks**
- Represent actual content, data or application functionality
- Encapsulated via **well-defined interfaces** (APIs)

User Roles

End users run Mashups

- Consume Mashup
- Personalize Mashup (e.g., change background color)

Key users create Mashups

- Add pre-build widgets (from the catalogue)
- Connecting gadgets by wiring their in-/out-ports

Consultants create Gadgets

- Binding generic User Interfaces to resources
- Transforming and aggregating data (piping)

Developers implement the services

- Create and deploy services
- Make resources available (e.g., RSS Feeds, SAP Enterprise Services)

Volker Hoyer. Enterprise Mashups

Mashups

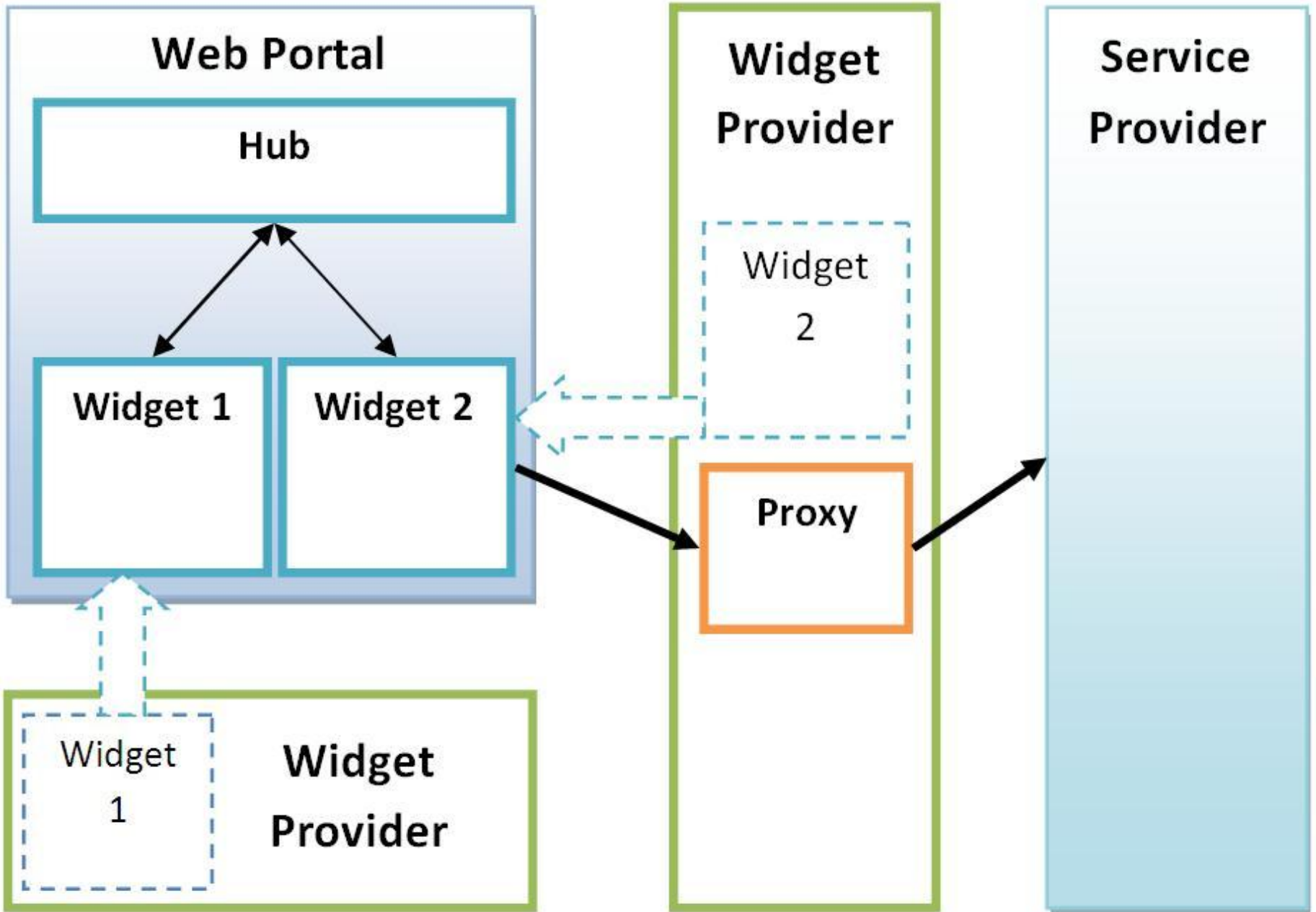
- Olemasolevad platvormid
- Yahoo! Pipes, IBM Lotus Mashups, Microsoft Popfly, Kapow, JackBe, Pesto, Netvibes, Intel Mash Maker
- Kõikidel omad probleemid

Hetkeprobleemid

- Enamus praeguseid vidinaid on üpris lihtsad
- Info kuvamine & lihtsam funktsionaalsus
- Vidinatevaheline suhtlus puudub
- Vidinatevaheline koostöö raskendatud
- Kolmandate osapoolte vidinate vaheline koostöö (peaaegu) võimatu

Tulevik ehk täna

- Nõuded keerulisematele mashup-rakendustele
 - Vidinatevaheline hea koostöö
 - Ressursside (vidinate) kombineerimine kolmandate osapoolte serveritest
 - Turvalisuse probleemid juhuslike vidinate kasutamisel
- Raamistik, mis seda kõike toetab
 - OpenAjax Hub



OpenAjax Hub

- Infrastruktuur, raamistik
- Keskne hub (switch)
- Sõnumivahetus vidinate vahel
- Turvalisus: vidinad Iframe konteinerites
- Turvapoliitika: millised vidinad tohivad omavahel suhelda

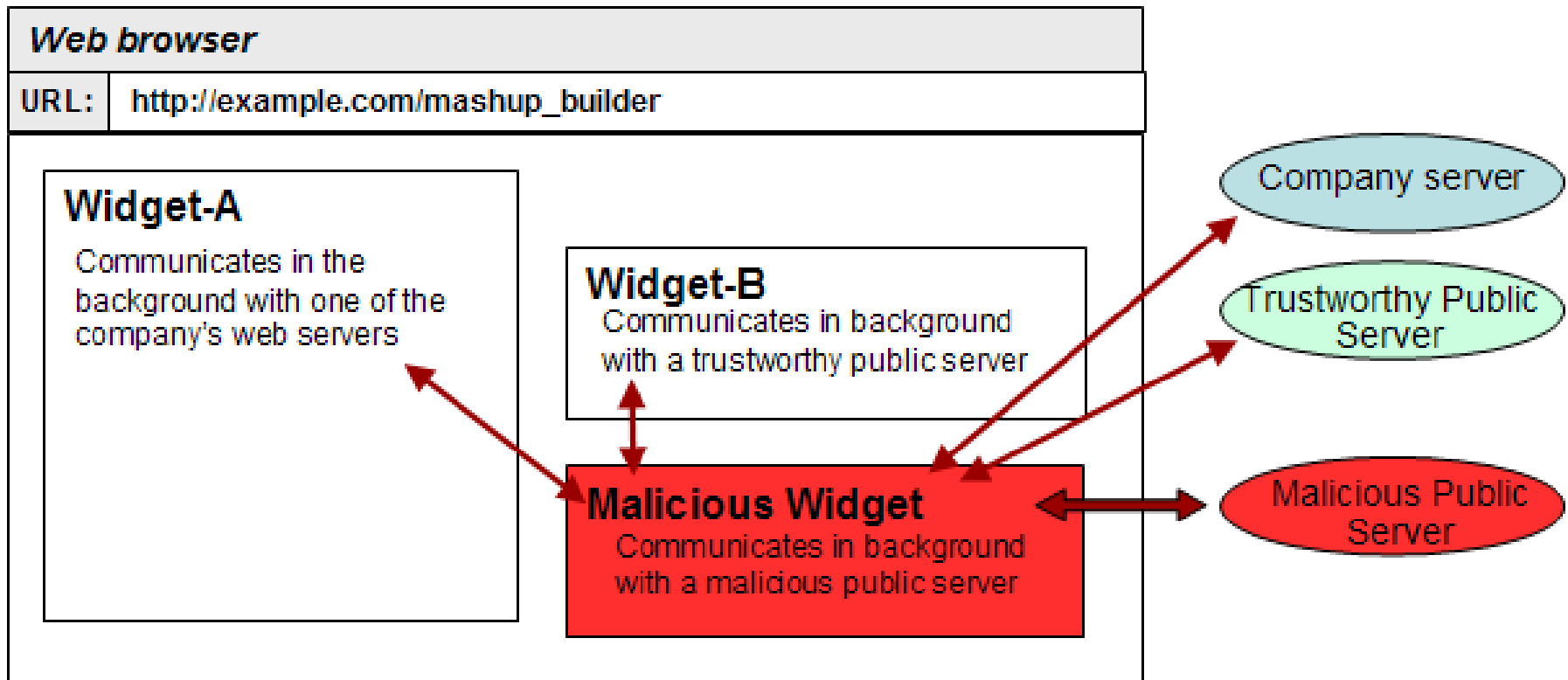
OpenAjax Hub

- Live Demo

OpenAjax Hub

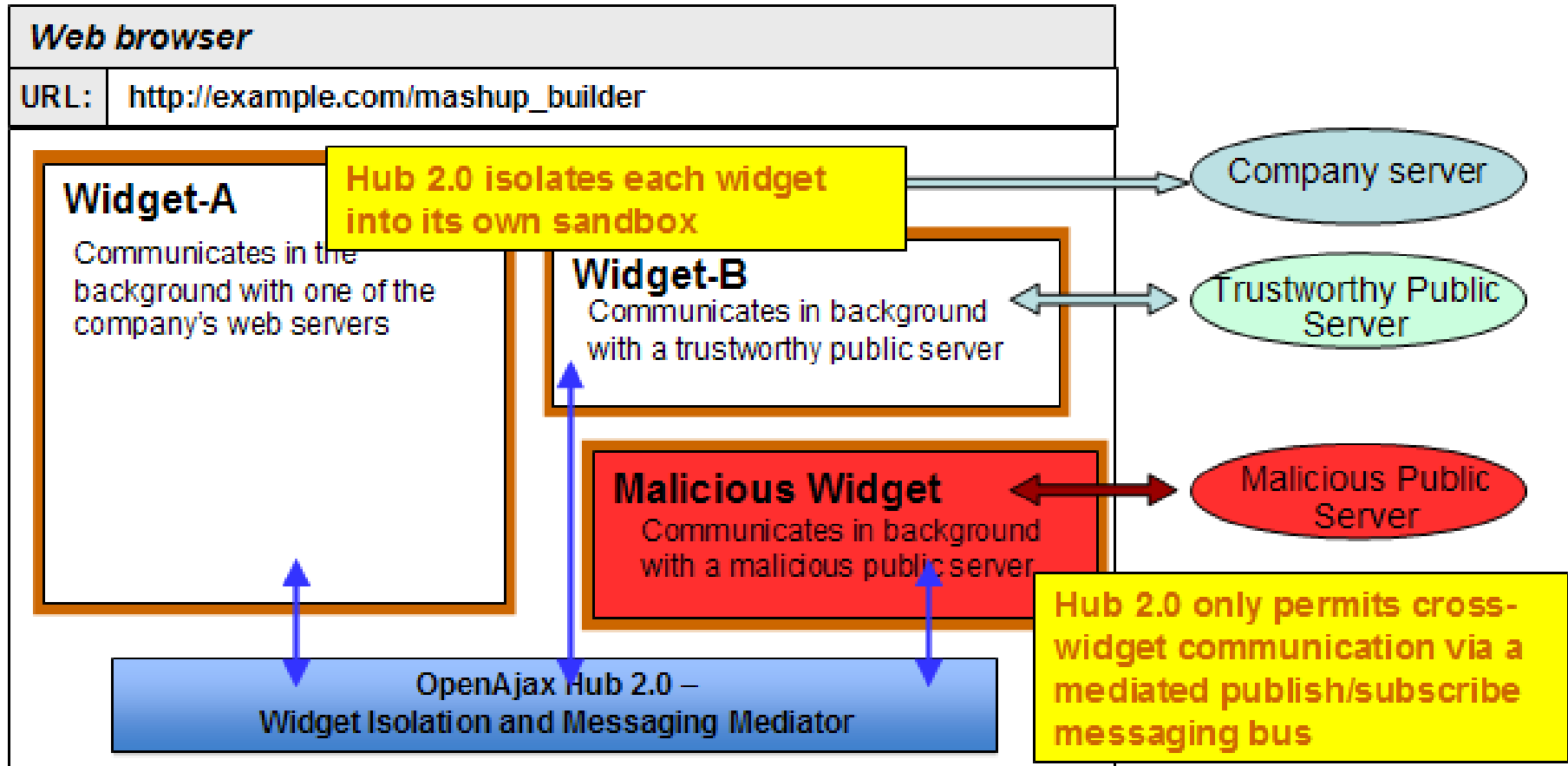
- Vidinad ei ole pelgalt info kuvamiseks
- Vidinaid saab panna omavahel koostööd tegema
- Infovahetus

Problem: turvalisus



OpenAjax Alliance

Turvalisus: lahendus



OpenAjax Alliance

OpenAjax Hub: Turvalisus

- Vidinad on isoleeritud **IFrame** konteineritesse
- Suhtlemine on lubatud ainult läbi Hub-i
- Hub-is on võimalik defineerida **turvapoliitikaid**
 - Millised vidinad tohivad **omavahel suhelda**
 - Millised vidinad tohivad sõnumeid **publitseerida**
 - Millised vidinad tohivad sõnumeid **kuulata**
 - Mida teha **turvamurrete** (security breach) korral

OpenAjax Hub

- Demo – veebilehele Hub-i lisamine

OpenAjax Hub - Veebilehele

- OpenAjax Hub-i uue instantsi loomine (JavaScript)

```
var managedHub = new OpenAjax.hub.ManagedHub({  
  onPublish:    onMHPublish,  
  onSubscribe:  onMHSubscribe,  
  onUnsubscribe: onMHUnsubscribe,  
  onSecurityAlert: onMHSecurityAlert  
});
```

OpenAjax Hub – Turvapoliitika Callback meetodid

//Luba kõikide sõnumite edastus

```
function onMHPublish(topic, data, publishContainer, subscribeContainer) {  
    return true  
}
```

//Luba kõikidel vidinatel tellida kõiki teemasid

```
function onMHSubscribe(topic, container) {  
    return true  
}
```

//Luba kõikidel vidinatel tellimused lõpetada

```
function onMHUnsubscribe(topic, container) {  
    return true  
}
```

//Tegele turvaprobleemidega

```
function onMHSecurityAlert(source, alertType) {  
}
```


OpenAjax Hub – Vidina lisamine

```
var mashupArea = document.getElementById("mashupArea");
var div1 = document.createElement( "div" );
mashupArea.appendChild(div1);
var container1 = new OpenAjax.hub.IframeContainer(managedHub , "client1", {
    Container: {
        onSecurityAlert: onClientSecurityAlert,
        onConnect:    onClientConnect,
        onDisconnect: onClientDisconnect
    },
    IframeContainer: {
        parent:    div1,
        iframeAttrs: { id: "firstcol" },
        uri: "http://www.example.com/widget_1.html",
        tunnelURI: "http://localhost:8080/test/tunnel.html"
    }
}
);
```

Konteineri meetodid

- `//Handle client-side security alerts`
- `function onClientSecurityAlert(source, alertType) {}`
- `//Called when client connects`
- `function onClientConnect(container) {}`
- `//Called when client disconnects`
- `function onClientDisconnect(container) {}`

Veebilehe näidis-HTML

```
<html>
<head>
  <script src="OpenAjaxManagedHub-all.js"></script>
  <script type="text/javascript">

    function onMHSecurityAlert(source, alertType) {
    }
    //Siia tulevad igasugu abimeetodid

    function loadEventHandler() {
      //Siia tuleb Hub-i ja vidinate initsialiseerimine
    }
  </script>
</head>
<body onload="loadEventHandler()">
  <div id="mashupArea"></div>
</body>
</html>
```

OpenAjax Hub – vajalikud failid

- **OpenAjaxManagedHub-all.js** – OpenAjax Hub-i JavaScripti teek
- **tunnel.html** – abifail vidinate ja Hub-i suhtlemise võimaldamiseks

OpenAjax Hub

- Vidinate veebilehele lisamise vahekokkuvõte
 - OpenAjax Hub-i JavaScripti **teek**
(OpenAjaxManagedHub-all.js)
 - **Turvapoliitika** callback meetodid
 - OpenAjax Hub-i **initsialiseerimine**
 - Vidinate lisamine (**konteinerite** loomine)
 - Vidinakonteinerite **abimeetodid**
 - body **onLoad meetod**, mis tõmbaks rakenduse käima

OpenAjax Hub – uue vidina loomine

- Demo - lähtekood

Vidin - initialiseerimine

```
function loadEventHandler() {  
    hubClient = new OpenAjax.hub.IframeHubClient({  
        HubClient: {  
            onSecurityAlert: clientSecurityAlertHandler  
        }  
    });  
  
    // Connect to the ManagedHub  
    hubClient.connect( connectCompleted );  
}
```

Vidin - abimeetodid

// Handle security alerts

```
function clientSecurityAlertHandler(source, alertType) {  
}
```

//Vidina klient, mis ühendub Hub-i külge ja mille kaudu saab

//Hub-iga ühendust pidada

```
OpenAjax.hub.IframeHubClient
```


Vidin - abimeetodid

```
//Kutsutakse peale Hub-iga ühenduse loomist
```

```
function connectCompleted ( hubClient, success, error ) {  
    if (success) {  
        hubClient.subscribe('org.example.topic', onData);  
    }  
}
```

```
//Kutsutakse välja iga kord,
```

```
//kui vidin saab sõnumi teemal 'org.example.topic'
```

```
function onData(topic, data) {  
    var messageArea = document.getElementById('messageArea');  
    messageArea.innerHTML = data;  
}
```

Vidin - abimeetodid

//publish (topic, data) – meetod sõnumite saatmiseks
hubClient.publish('org.some.topic','Hello World');

//subscribe (topic, callback_method) – teemade tellimine
hubClient.subscribe('org.example.topic', onData);

Vidin – näidis-HTML

```
<html>
<head>
  <script type='text/javascript' src="OpenAjaxManagedHub-all.js"></script>
  <script type='text/javascript'>
    function clientSecurityAlertHandler(source, alertType) {}
    //Ülejäänud abimeetodid

    function loadEventHandler() {
      //Initsialiseerida vidina klient ja ühendada Hub-iga
    }
    function clickEvent() {
      hubClient.publish('org.some.topic','Hello World');
    }
  </script>
</head>
<body onload="loadEventHandler();">
  <div>
    <input type="button" value="Send Message" onclick="clickEvent()"/><br />
    <div id="messageArea"></div>
  </div>
</body>
</html>
```

Uue vidina loomise vahekokkuvõte

- Luua **IframeHubClient** instants
- Ühenduda Hub-i külge (**connect**)
- Callback meetod, kui Hub-iga **ühendus valmis**
- **Subscribe** soovitud teemade külge
- Callback meetod **saabunud sõnumite** töötlemiseks
- **Publish** – saada sõnumeid teistele vidinatele
- HTML lehel – initsialiseerida JavaScript loogika

OpenAjax Hub

- Koodinäiteid
- http://www.openajax.org/member/wiki/OpenAjax_Hub_2.0_Specification_Managed_Hub_Overview
- Spetsifikatsioon
- http://www.openajax.org/member/wiki/OpenAjax_Hub_2.0_Specification#Table_of_Contents
- Lähtekood
- <http://stacc.ee/~villido/ExampleMashup/ExampleMashup.zip>

Vahekokkuvõte

- Mis on vidinad (widgets)?
- Miks on vidinad kasulikud?
- Mis on mashup?
- Miks on mashupid kasulikud?
- Mis on OpenAjax Hub?

OpenAjax Hub Sõnumid

`//publish (topic, data) – meetod sõnumite saatmiseks`
`hubClient.publish('org.some.topic','Hello World');`

- Sõnumid võivad olla kõik JavaScripti objektid
 - String, näiteks “Hello World”;
 - Number, Boolean – 123.32 või true
 - JSON, näiteks
 - {message:”Hello World”, randomNumber:123.32}

JSON näide

```
{
  firstName: "John",
  lastName: "Smith",
  age: 25,
  address: {
    streetAddress: "21 2nd Street",
    city: "New York",
    state: "NY",
    postalCode: "10021"
  },
  phoneNumber: [
    { type: "home", number: "212 555-1234" },
    { type: "fax", number: "646 555-4567" }
  ]
}
```


Kui vidinad kasutavad erinevaid andmestruktuure?

Topic: org.geoinfo.points.coordinates

```
{location:{  
  coordinates:[  
    {latitude:58.36611, longitude:26.73611},  
    {latitude:59.36611, longitude:27.23611},  
    {latitude:60.36611, longitude:27.73611}  
  ]}  
}
```

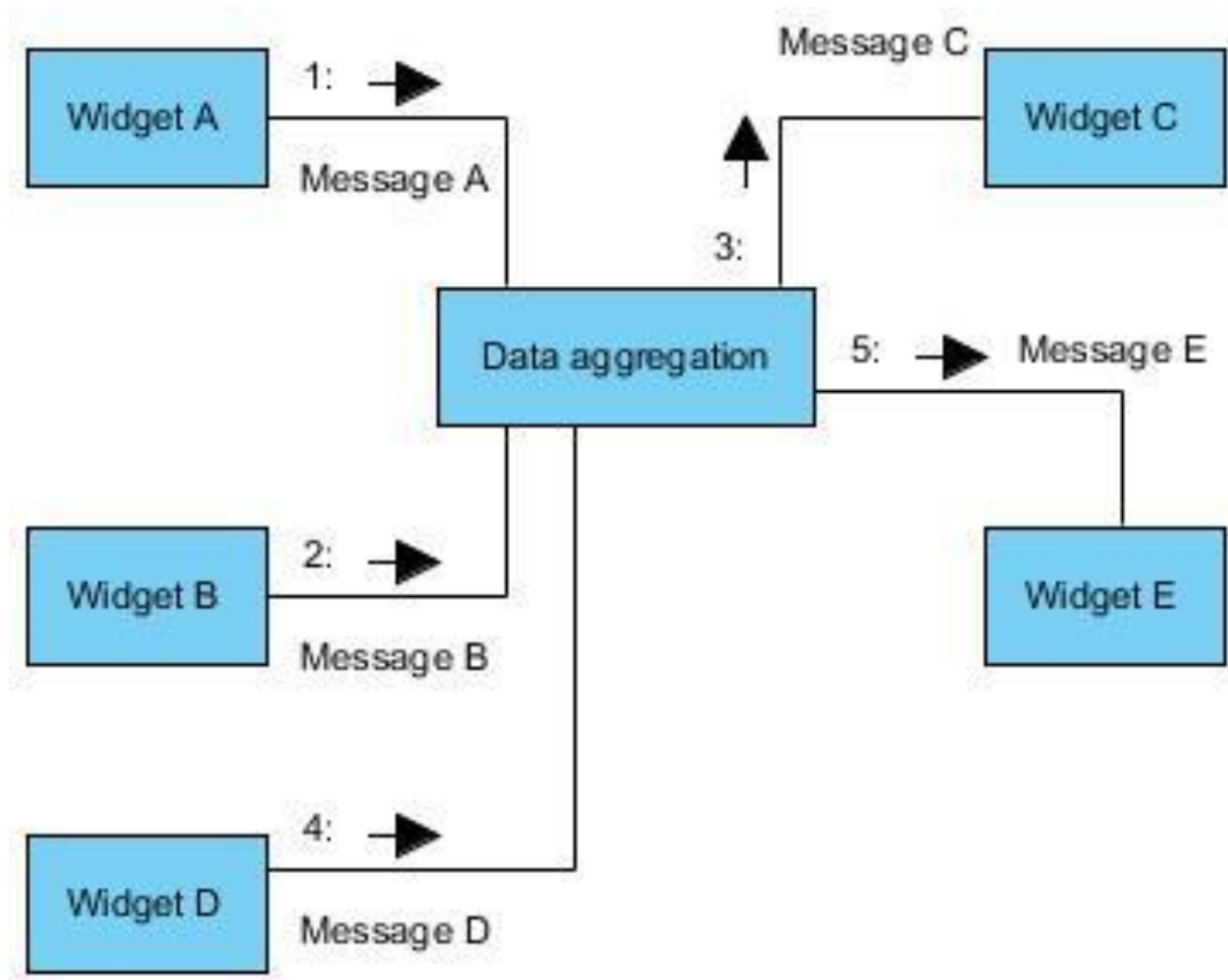
Topic: org.map.places

```
{places:[  
  {lat:58.36611, long:26.73611},  
  {lat:59.36611, long:27.23611},  
  {lat:60.36611, long:27.73611}  
]  
}
```

Vajadus sõnumeid tõlkida

- Vidinatel erinevad loojad
- Vidinad räägivad erinevat “keelt” st andmestruktuurid on erinevad
- Vahel vaja kombineerida sõnumeid erinevatest allikatest

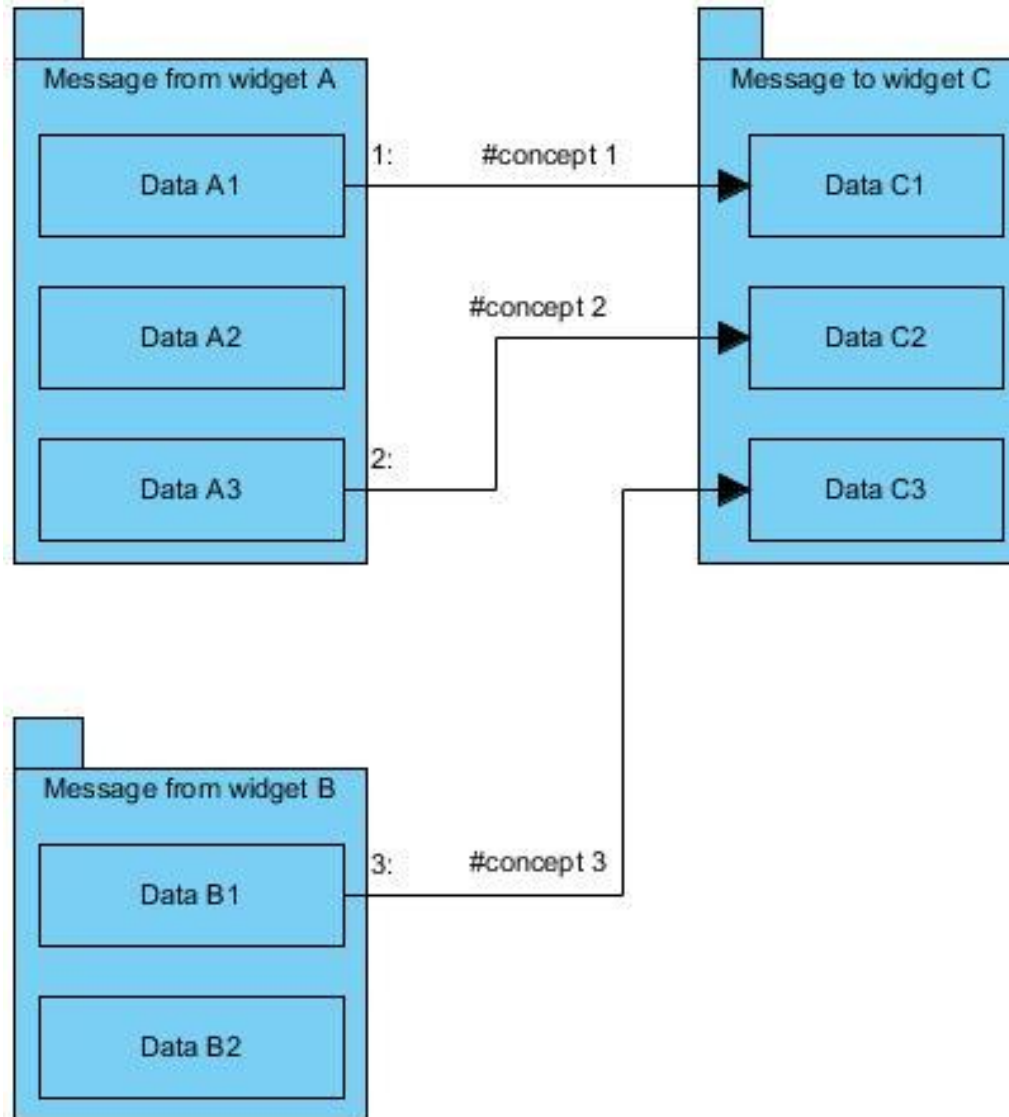
Sõnumite agregeerimine



Sõnumite agregeerimine

- Kõikidest sõnumitest kogutakse vajalikke andmeid
- Kui on kogunenud piisavalt andmeid, et mõnele vidinale uus sõnum saata, siis genereeritakse nendest andmetest uus sõnum ja saadetakse sellele vidinale

Uute sõnumite genereerimine



Sõnumite semantiline integratsioon

- Sõnumite sisu tuleb teha “masinale” arusaadavaks
- Andmeelemendid tuleb sõnumi seest kätte saada ja paigutada teiste sõnumite külge
- Sõnumi struktuur ja semantika tuleb muuta arusaadavaks

Sõnumite semantiline integratsioon

Mõlemad sõnumid kannavad informatsiooni koordinaatide latitude ja longitude kohta.

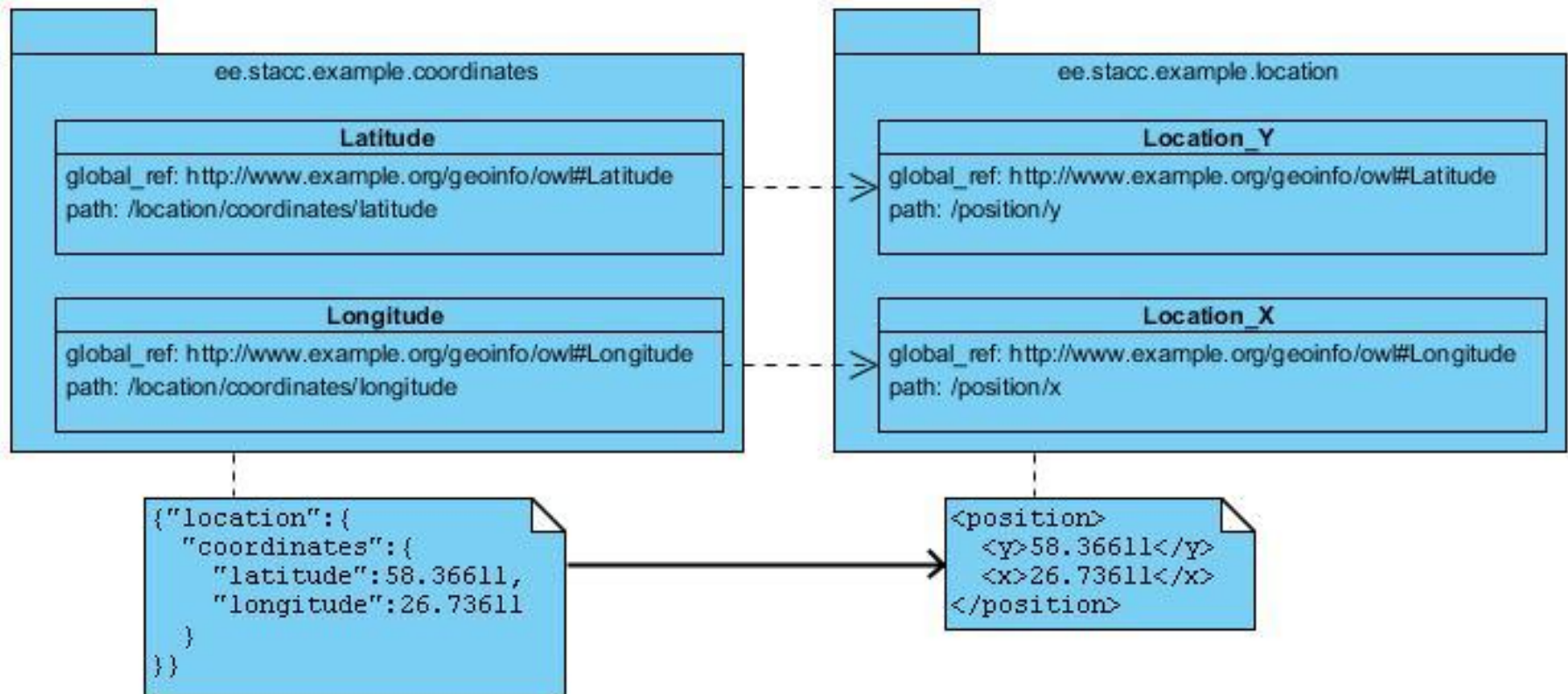
Topic: org.geoinfo.point

```
{location:{  
  coordinates:{  
    latitude:58.36611,  
    longitude:26.73611  
  }  
}}
```

Topic: org.map.place

```
{place: {lat:58.36611, long:26.73611}}
```

Semantiline integratsioon



Path & OWL

Sõnum 1

```
{location:{  
  coordinates:{  
    latitude:58.36611, longitude:26.73611}  
  }  
}
```

Path: /location/coordinates/**latitude**

OWL: #Latitude

Sõnum 2

```
{place: {lat:58.36611, long:26.73611}}
```

Path: /place/**lat**

OWL: #Latitude

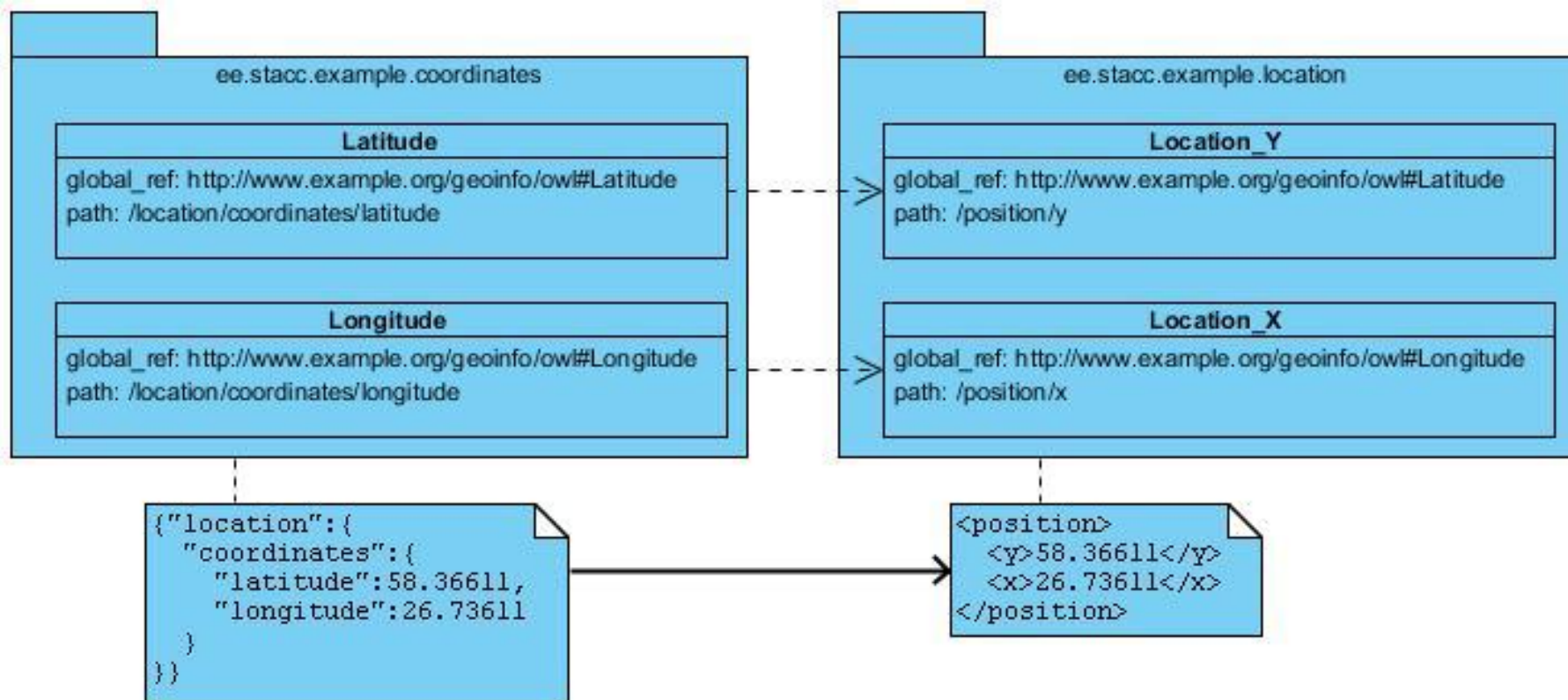
Path & OWL

- **Path** viitab andmeelemendi asukohale sõnumis
- **Path:** /location/coordinates/**latitude**
- **OWL** defineerib andmeelemendi olemuse
- Viide ontoloogia mõistele (global_ref)
- <http://www.example.org/geoinfo#Latitude>

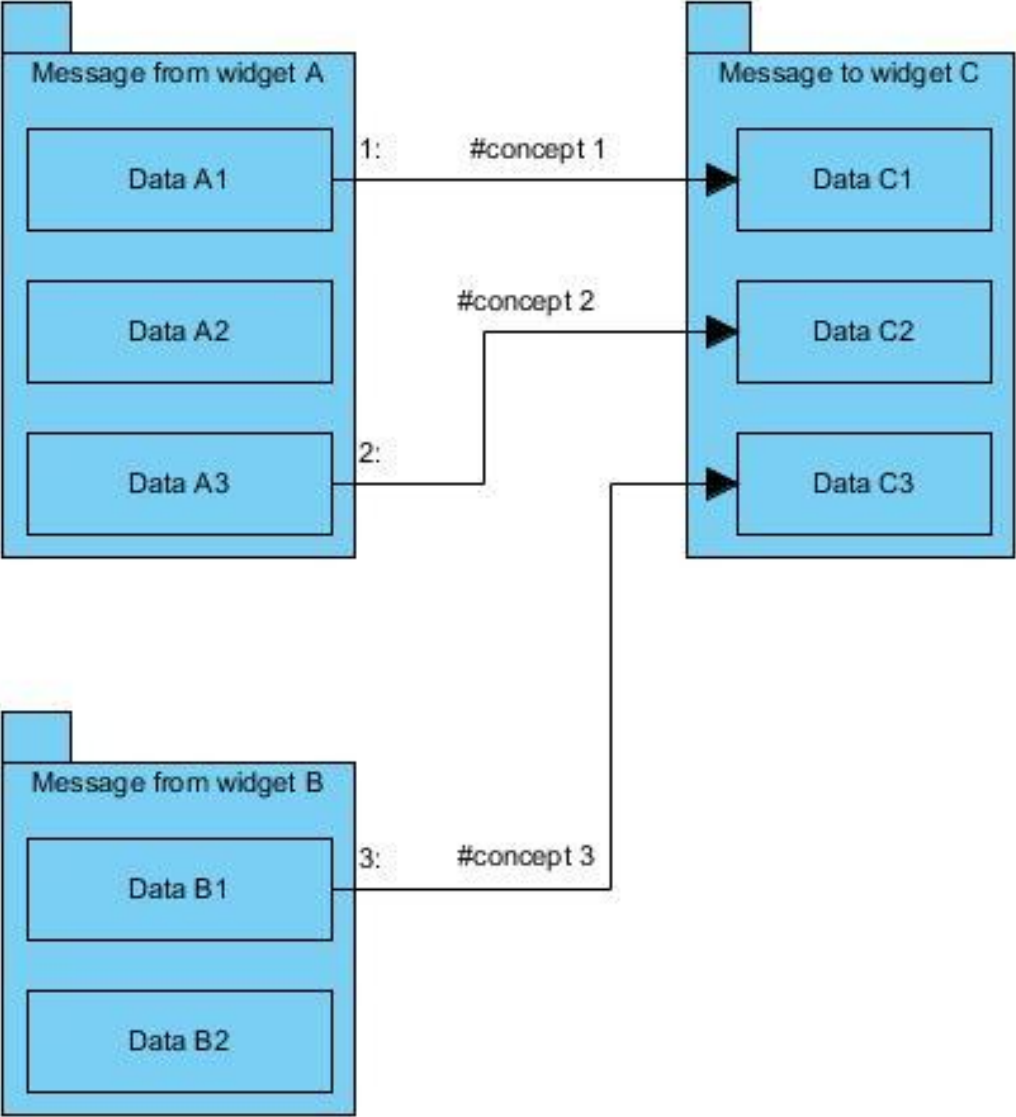
Path & OWL

- **Path** ja **OWL** (global_ref) on piisavad, et kirjeldada sõnumi sisu masinloetavaks
- **Path**-i ja **OWL**-ga saab kõik sõnumid ära kirjeldada
- Kui kahes sõnumis on andmeelement, mis viitab samale **OWL**-i mõistele, siis järelikult kannavad need andmeelemendid samasuguseid andmeid

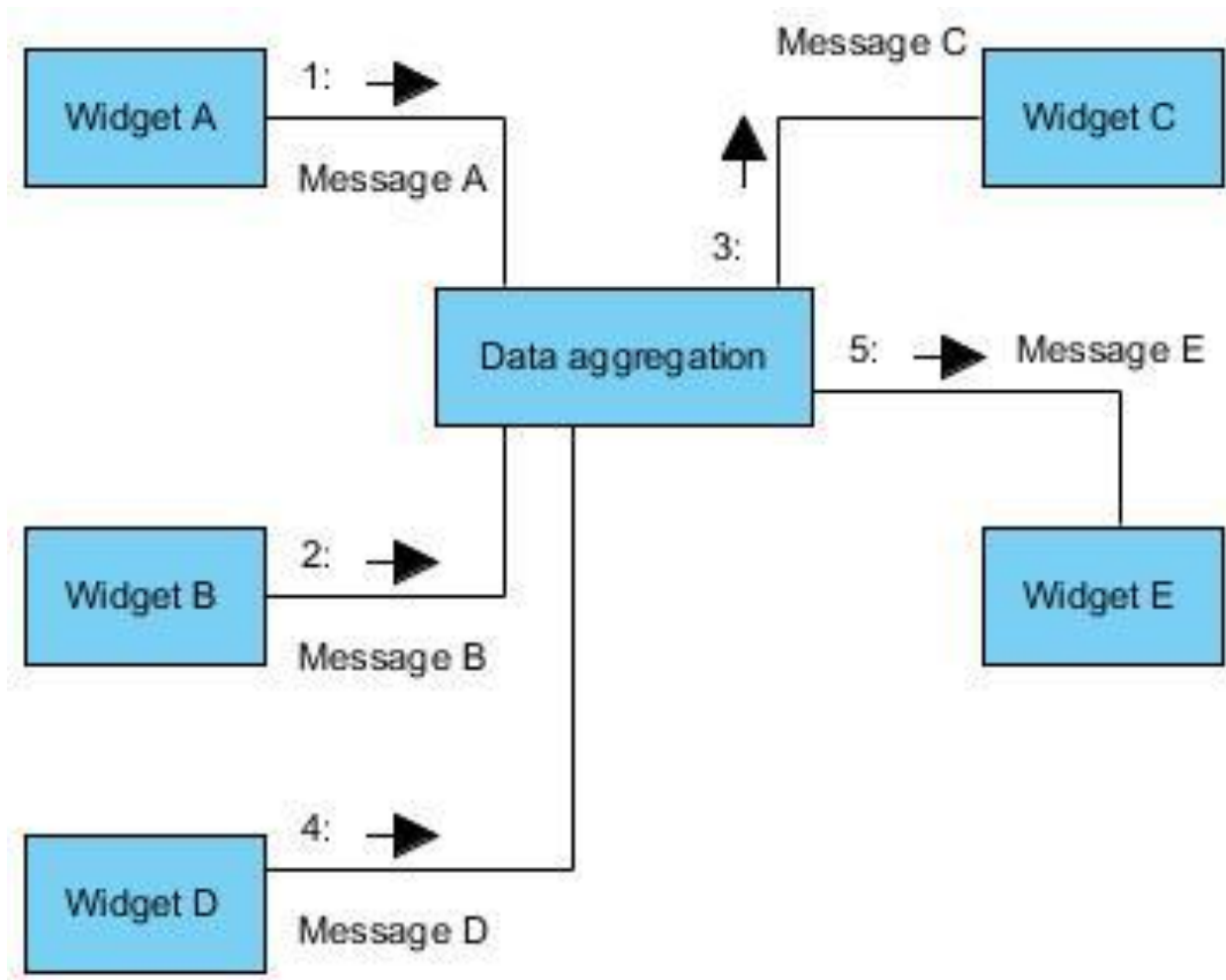
Semantiline integratsioon



Uute sõnumite genereerimine



Sõnumite agregeerimine



Sõnumite Mappingud

```
<frame>
  <topic>org.geoinfo.point</topic>
  <format>json</format>
  <schema>schemas/timespace.js</schema>
  <mappings>
    <mapping>
      <global_ref>http://www.example.org/geoinfo#Latitude</global_ref>
      <path>/location/coordinates/latitude</path>
    </mapping>
    <mapping>
      <global_ref>http://www.example.org/geoinfo#Longitude</global_ref>
      <path>/location/coordinates/longitude</path>
    </mapping>
  </mappings>
</frame>
```

JSON Schema

```
{ "type": "object",  
  "properties": {  
    "location": {  
      "type": "object",  
      "properties": {  
        "coordinates": {  
          "type": "object",  
          "properties": {  
            "latitude": { "type": "number" },  
            "longitude": { "type": "number" }  
          }  
        }  
      }  
    }  
  }  
}
```

Sõnum

```
{ location: {  
  coordinates: {  
    latitude: 58.36611,  
    longitude: 26.73611  
  }  
}}
```


Sõnumite Mappingud

- mappings.xml
- Demo

Mappingud – korduvad elemendid (massiivid)

```
{places:[  
  {lat:58.36611, long:26.73611, srs:"EPSG:4326"},  
  {lat:59.36611, long:27.23611, srs:"EPSG:4326"},  
  {lat:60.36611, long:27.73611, srs:"EPSG:4326"}  
]  
}
```

Mappingud – korduvad elemendid

```
<frame>
  <topic>org.places.list</topic>
  <format>json</format>
  <schema>schemas/placesList.js</schema>
  <mappings>
    <repeating_element_group path="/route/places">
      <mapping
        global_ref="http://www.example.org/geoinfo/#Latitude"
        path="/places/lat" />
      <mapping
        global_ref="http://www.example.org/geoinfo/#Longitude"
        path="/places/long" />
      <constant path="/route/places/srs" value="EPSG:4326" />
    </repeating_element_group>
  </mappings>
</frame>
```

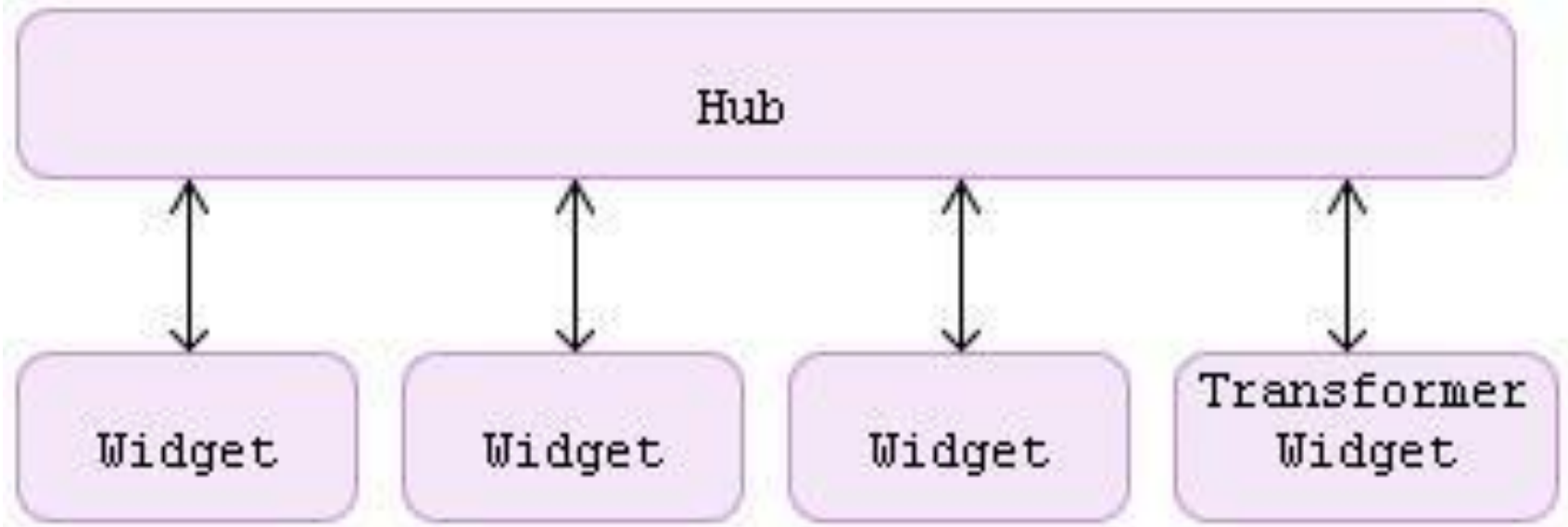
JSON Schema - massiivid

```
{  
  "type":"object",  
  "properties":{  
    "places":{  
      "type":"array",  
      "items":{  
        "type":"object",  
        "properties":{  
          "lat":{"type":"number"},  
          "long":{"type":"number"},  
          "srs":{"type":"string"}  
        }  
      }  
    }  
  }  
}
```

Transformer Widget (Transformaator)

- Transformaator (Transformer Widget)
- Mappingute põhjal sõnumite transformatsioonide võimaldamiseks
- Vaja lisada OpenAjax Hub-i külge nagu tavaline vidin

Transformer Widget



Transformer Widget

- Demo
- Transformaatori lisamine

Transformer Widget

- Lisatakse Hub-i külge nagu tavaline vidin
- **mappings.xml** – defineeritakse sõnumite mappingud
- **Schemas** kaust – JSON-i skeemid
- transformerwidget kataloog – Transformaatori failid
- TransformerWidget.html – Transformaatori laadimiseks

Kokkuvõte

- Miks on transformatsioonid vajalikud?
- Mis on Transformaator (Transformer Widget)?

Küsimusi?