

# Data Access Management Pattern

Sergey Skudarnov

Saint-Petersburg State University

[sergey.skudarnov@gmail.com](mailto:sergey.skudarnov@gmail.com)

# Preface

- The topic based on Ubiq Mobile project – a universal platform for mobile online services
- DAMP system uses in Ubiq Mobile server as a module for interaction between applications and external data sources

# The Problem

- Applications deployed on server needs to work with external data
- Uniform interaction model
- Wide range of potentially supported data sources
- Easy development of applications

# Our Solution – DAMP


- **Data access unification**
- **Data access management**
- **Data abstraction and manipulation**
- **Wide range of supported data sources**
  - HTML , XML, files, databases, RSS feeds, Web services, TCP/IP, etc.

# DAMP Targeting



**Specialized Frameworks**

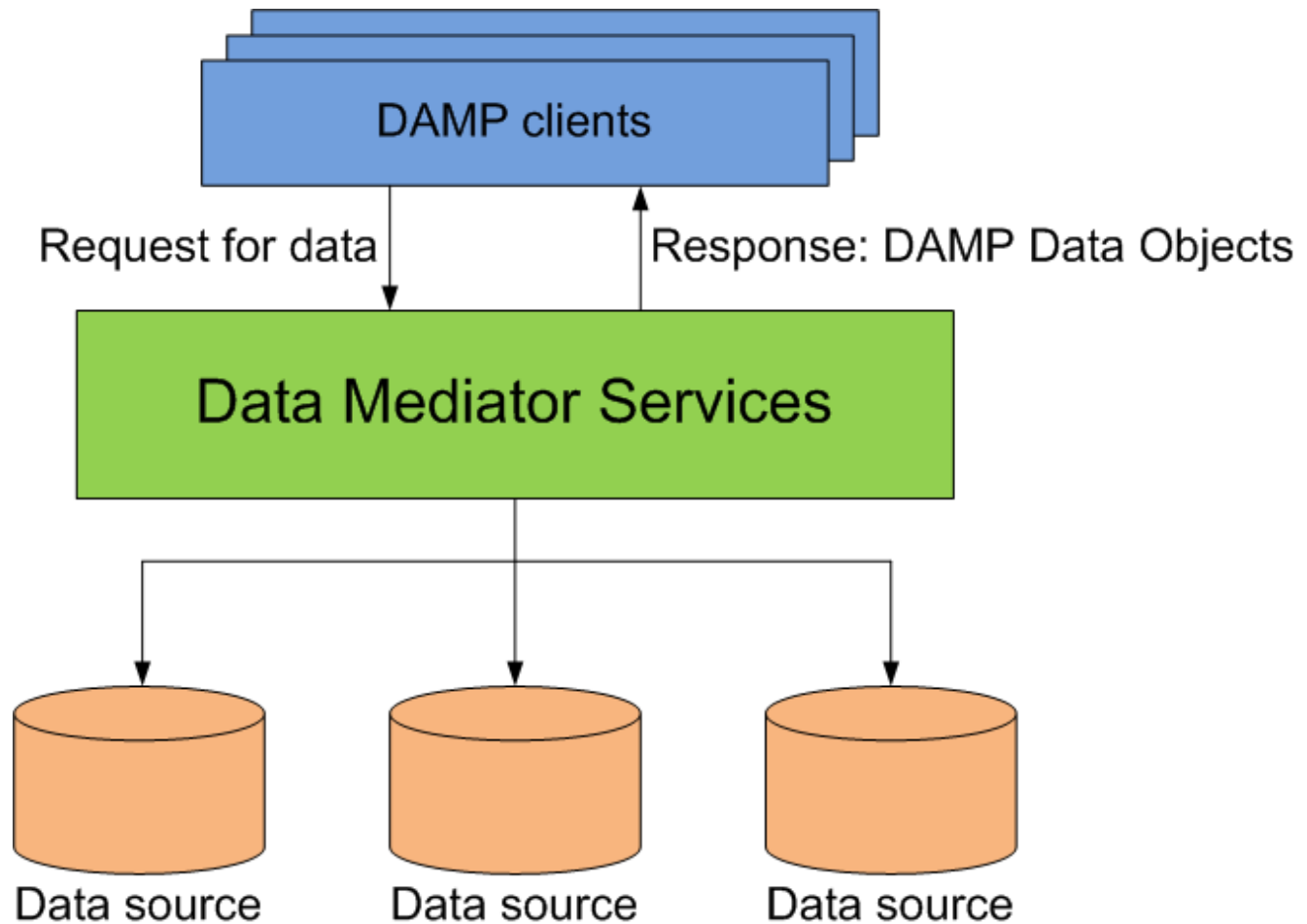
- + rich functionality
- heavyweight
- limited set of supported data sources
- platform-dependency...

- 
- limited functionality (e.g. Open, Close, Read, Write, Seek, etc.)
  - + unification
  - + lightweight
  - + wide range of potentially supported data sources...

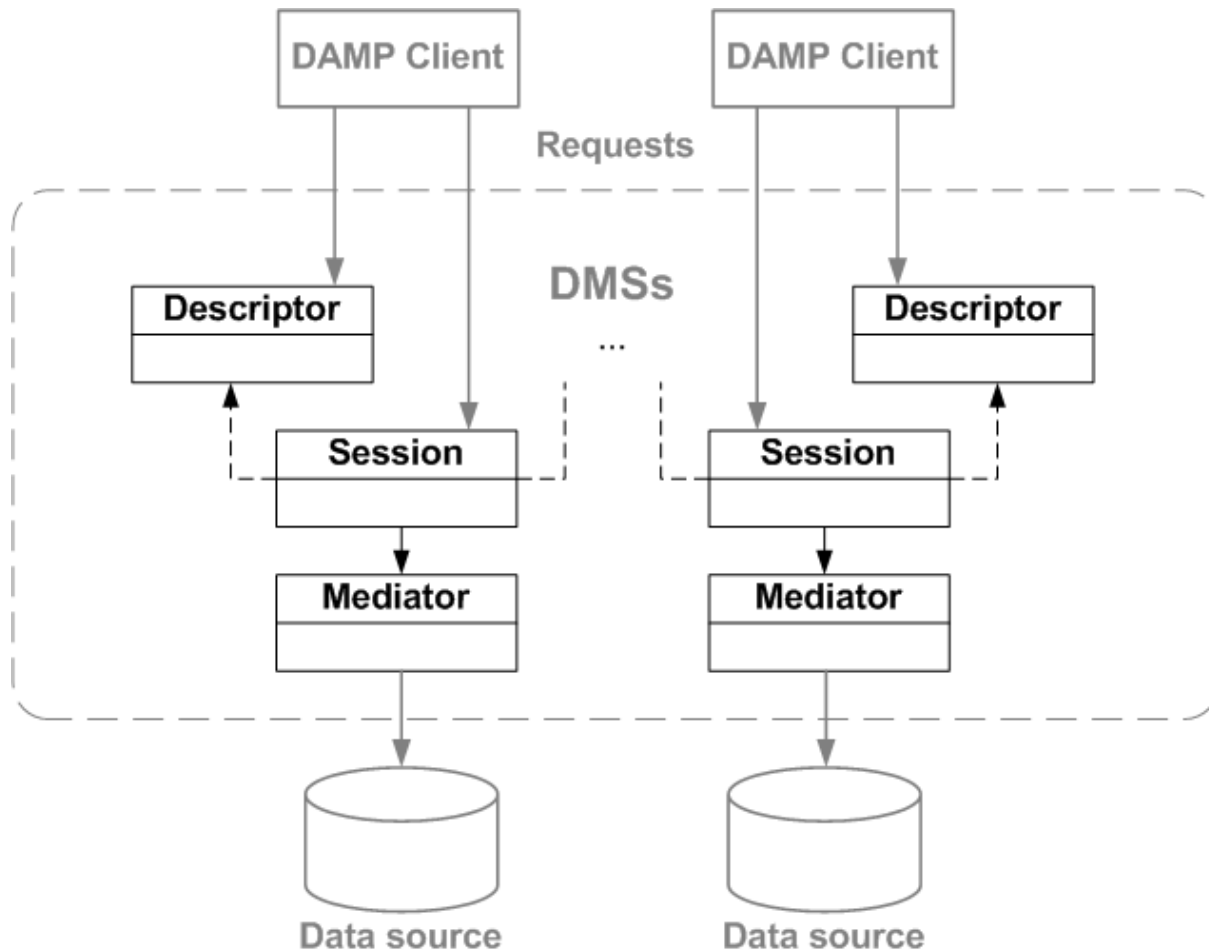


**DAMP**

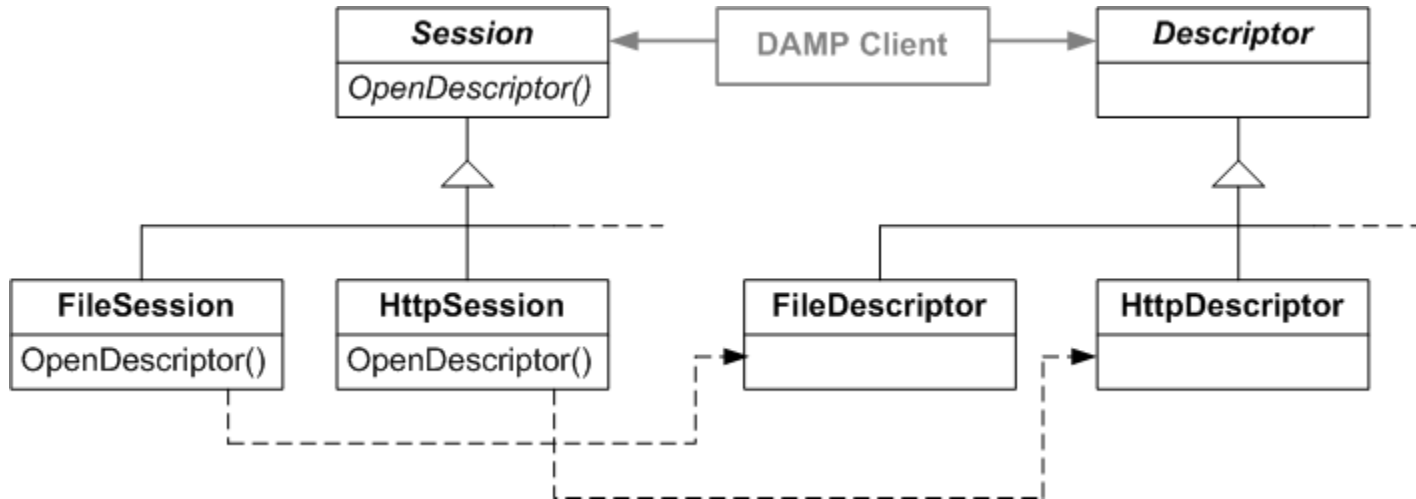
# General Architecture



# Data Mediator Services



# Descriptors & Sessions





# DAMP Data Objects

- **RawSessionData (RSD)**
  - intelligent buffers for data retrieved via Mediators
  - managed by Descriptors
- **SessionData**
  - customized object-oriented representation of raw data
  - could be predefined (by DAMP system) and client-defined

# SessionData

- Client-defined SessionData's structure depends only on client's needs
- DMSs don't know how to map raw data to client's representation
- SessionData objects has special method FillContent() that implements mapping process (implementing by clients)



# Sample Code:

## Predefined SessionData usage

```
FileSession fileSession;  
TextFile* textFile =  
    fileSession.OpenTextFile("example.txt",  
                             ERead);  
  
// textFile usage...  
delete textFile;
```

# Sample Code:

## Client-defined SessionData usage [1/2]

```
class MyHtml : public SessionData {
public:
    virtual void FillContent(RawSessionData&);
    // ...
};

void MyHtml::FillContent(RawSessionData&
                        rawSessionData) {
    RawHtmlData& rawData =
        static_cast<RawHtmlData&>(rawSessionData);
    // further implementation...
}
```

# Sample Code:

## Client-defined SessionData usage [2/2]

```
MyHtml myHtml;  
// ...  
HttpSession httpSession;  
HttpDescriptor* descr =  
    httpSession.OpenHttp("http://...",  
                           ERead);  
descr->GetData(myHtml);  
httpSession.CloseHttp(descr);  
// myHtml usage...
```

# Future Activities

- Complete support of write operations
- Expansion of set of supported types of data sources