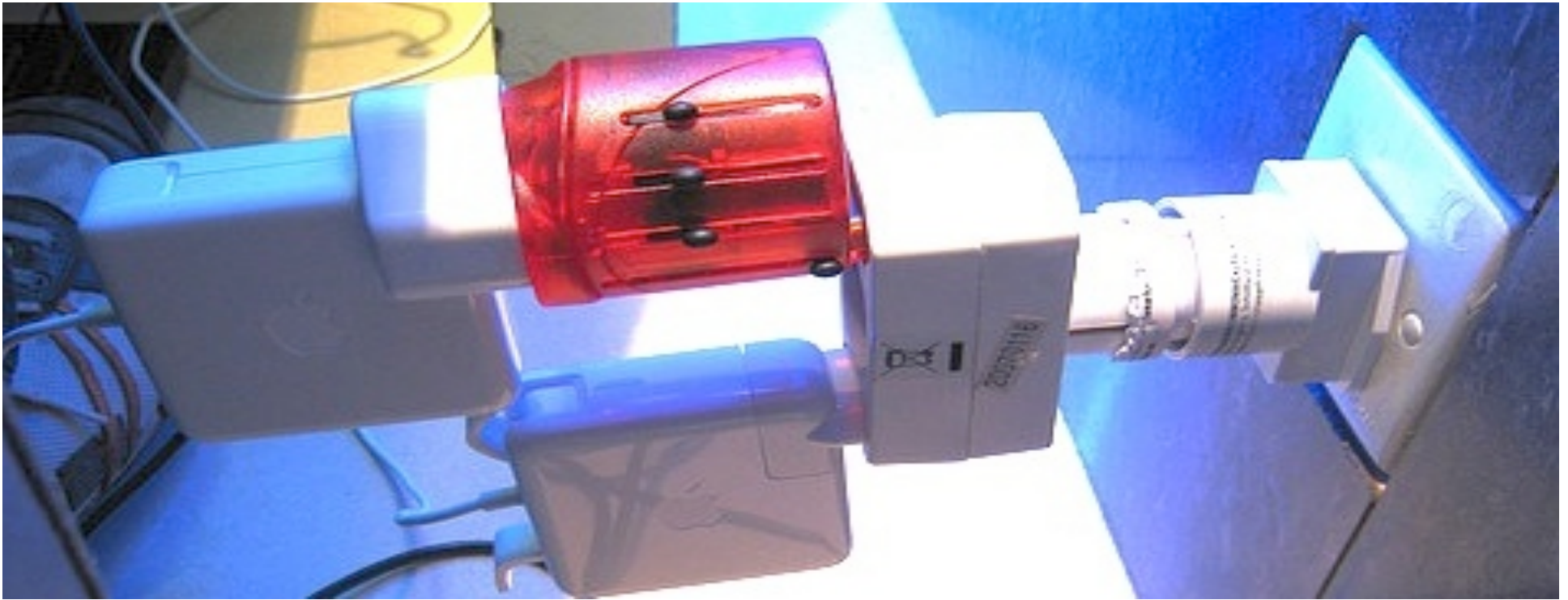




Interface Horror

Issues of cooperating systems

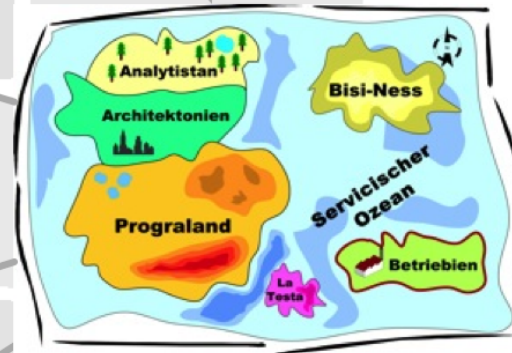


Follow @gernotstarke

innoQ
Fellow



aktiv



(co) founder of:



member of:



innoQ

Fellow

innoQ

Why talk / think about „interfaces“?

Why?

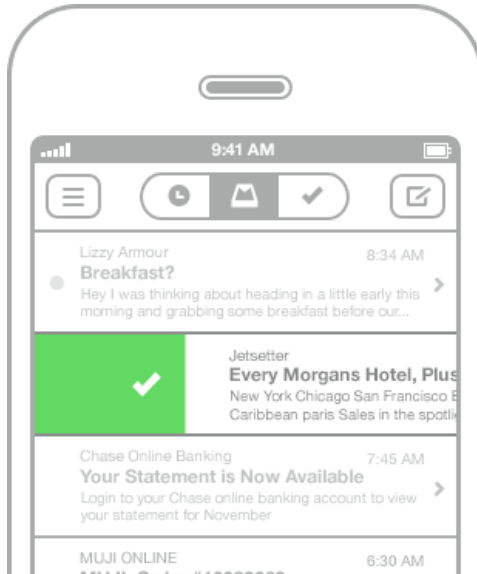
Systems **fail**
at interfaces...



http://www.city-data.com/forum/attachments/house/52333d1257441724-roof-wall-flashing-gutter-interface-problem-gutter_above.jpg

100.000.000 \$

price of good interface



Founded 2010.

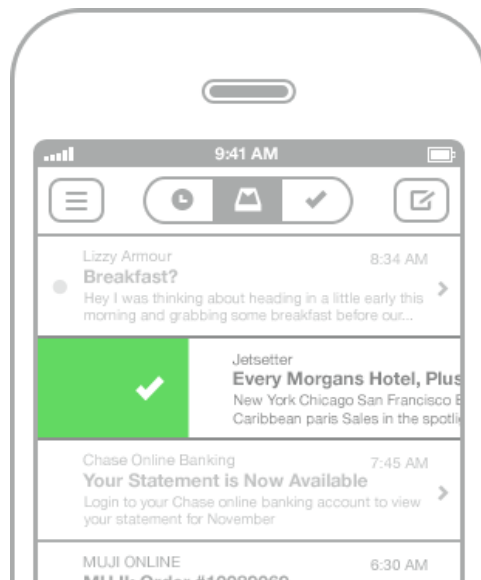


**New „interface“ to Email.
13 employees.**

Dropbox paid approx. 100 Million \$



100.000.000 \$

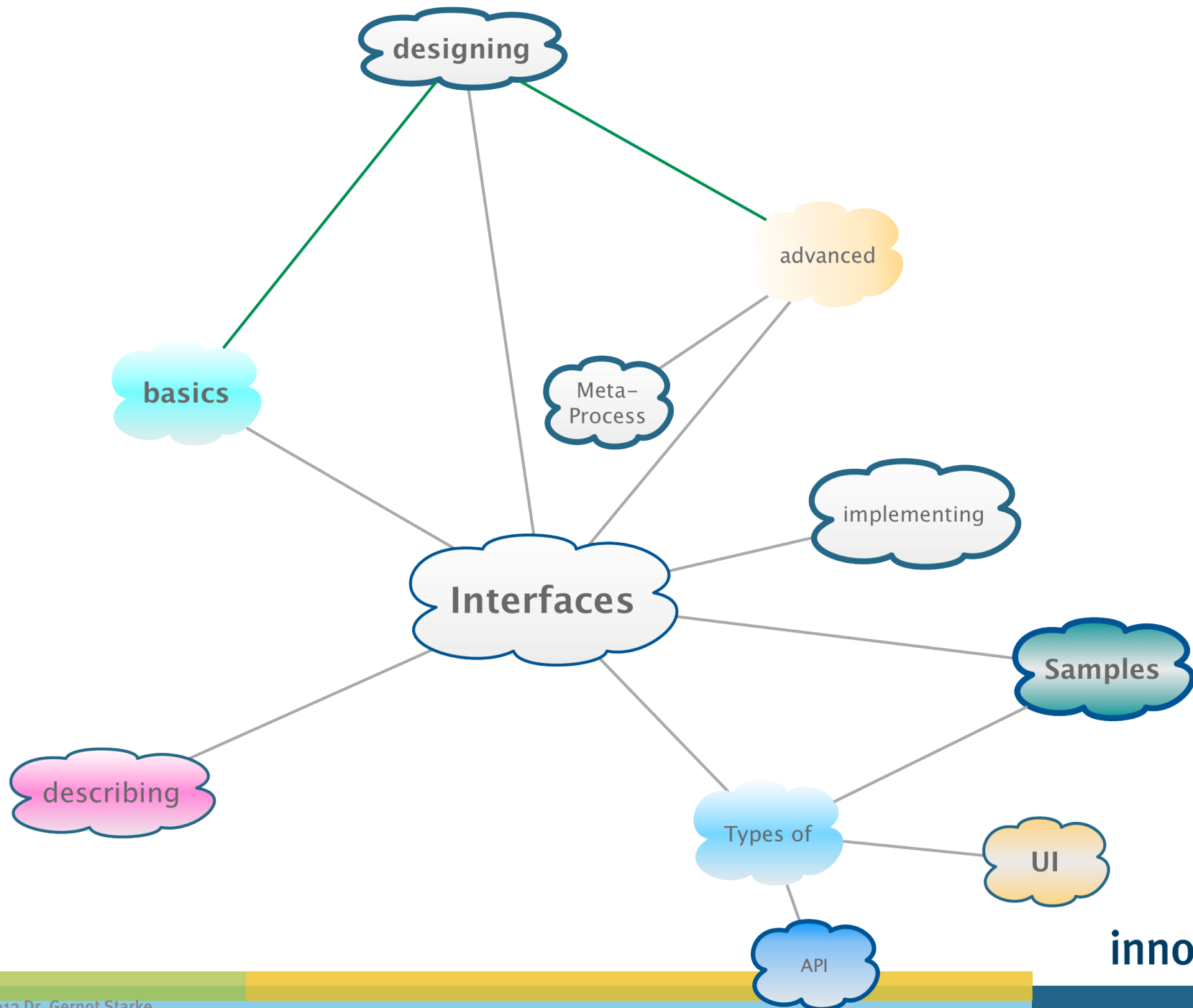


Why?

Systems **win** at interfaces...



What to discuss about „interfaces“?



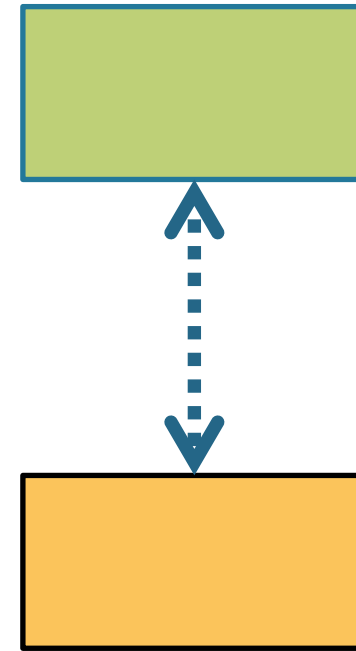
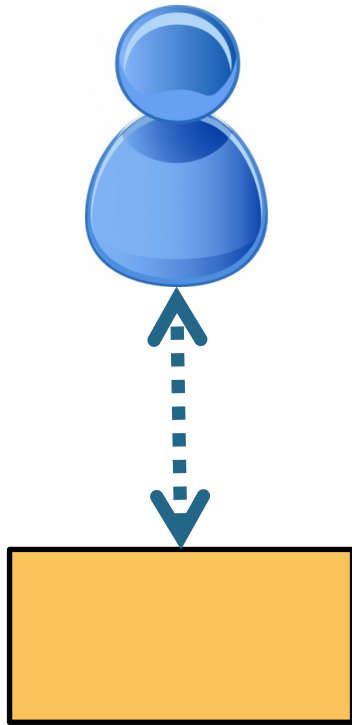


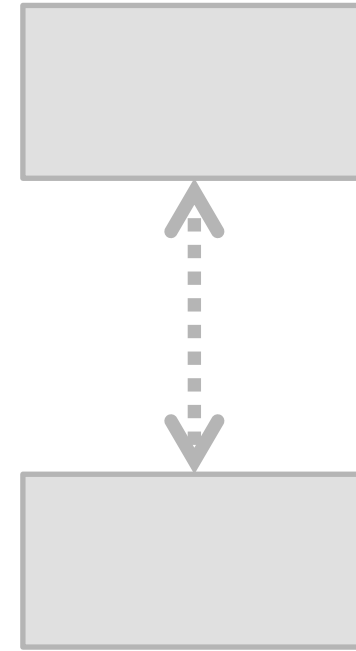
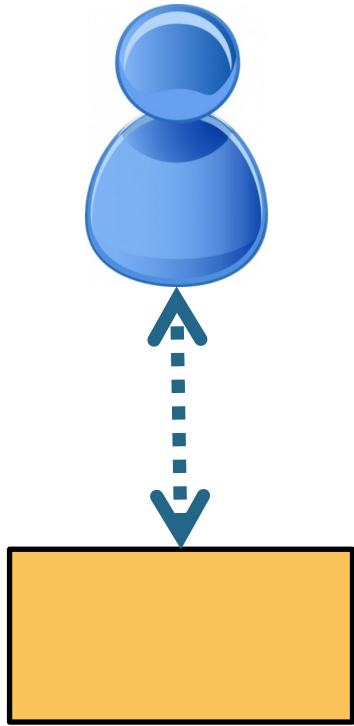
What's an „Interface“



Inter [latin]:
„Between“

Interface: between what?





User Interfaces



User Interface

Formatting

Font

Color

Ergonomics

Layout

Interaction

Language

Size

Psycho & Soft Issues

Visual Style

MVC MVP Observer Binding UI-Framework

Swing JSP JavaFX Griffon Widgets HTML CSS

innoQ



Golden Rules for **Bad** User Interfaces...



- Make usage a challenge
- Make it **mouse-only**
- **Hide** important functions
- Help them in technical language



BAD User Interface

Fac: Order# 99004234 99031927

OCB SSF Dupe View Routing Print Call Canceled

Mode From SC To SC Find CA# 100670861

Charges: 761.50

Discount: 0%

SubTotal: 761.50

Accessorial: 40.00

DV: 0 0.00

FSC: CAZ 2.50% 38.00

Total: 839.50

Balance: 839.50

ADDEND

CLOSED

POST

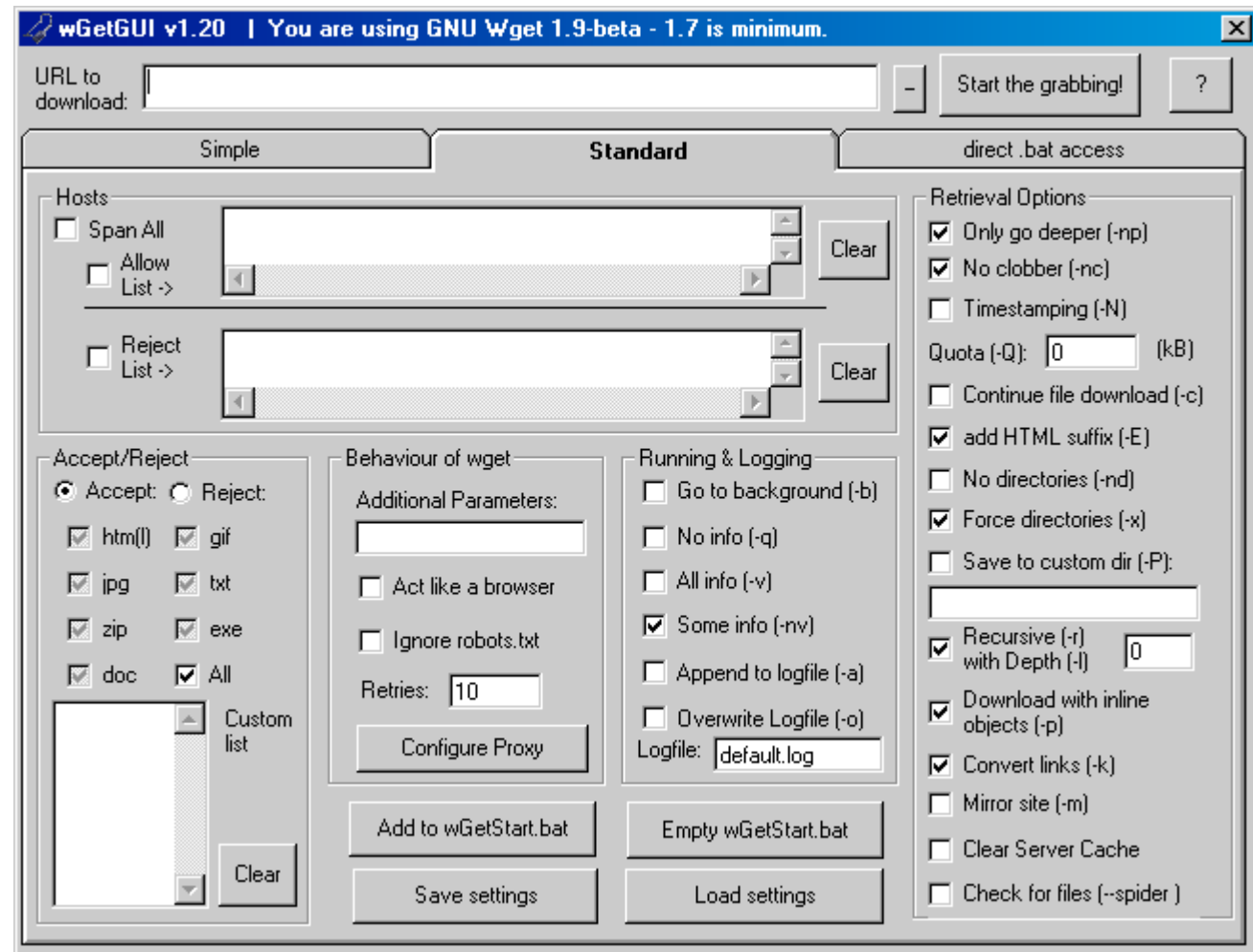
SAVED

Units	Type	H	Description	Stated	ActWT	Dimensions	Qty	Rate	Charge
1	CRATE		CRATE	91	94	97 25x54x30	97	50.00	48.50
1	2MAN		2 MAN P&D					40.00	40.00
2	CRATE		CRATE	500		1,426 50x49x49	1,426	50.00	713.00
0								0.00	0.00

Accs: 640.00 DV: 0 50.00 591 94 1523 1,523 761.50

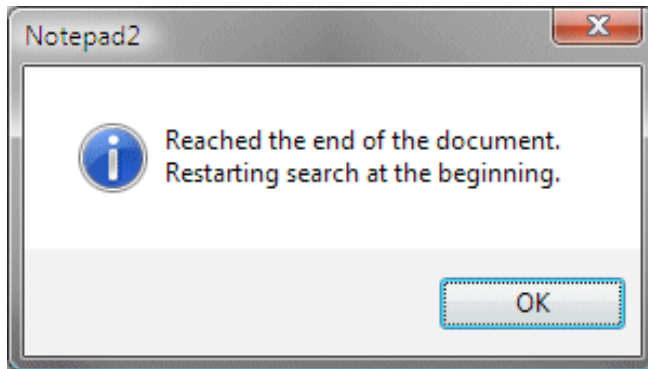
... By expert programmer 😊

Underlying data structure defines UI...

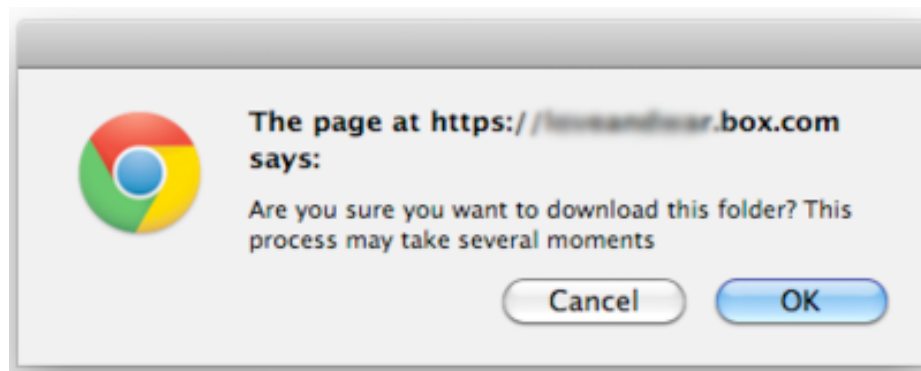




Golden Rules for Bad User Interfaces...



- **Block** user input often
- Let application **freeze** from time to time



Windows

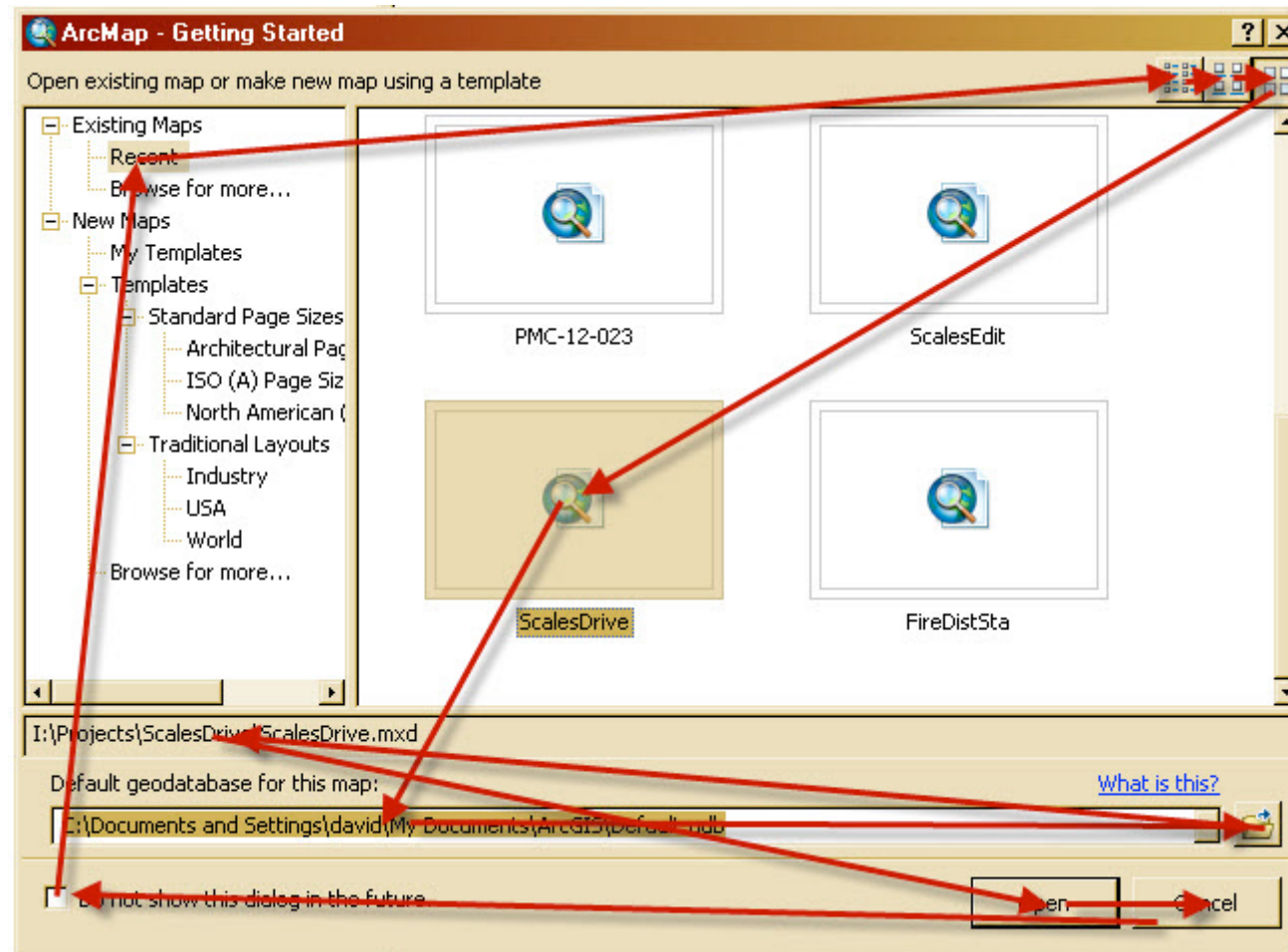
Windows crashed again. I am the Blue Screen of Death. No one hears your screams.

- * Press any key to terminate the application.
- * Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved data in all applications.

Press any key to continue _



TAB order madness ...





File Utility

File Actions Options Help

100923-Chandigarh-1
100927-Kharar
101007-Kharar-Haridv
101015-Jaipur-Jaisalr
101019-Jaisalmer-Jod
101031-Udaipur-Goa
101102-Goa
dia-abend
dia-abend.blog
Für-das-Bloo

Name	New Name	Sub...	Type	Size	Created	Mo
01-delhi-erstes-photo.JPG	01-delhi-erstes-photo.JPG		JPG F...	2 MB	21.11...	11.
02-delhi-imbau.JPG	02-delhi-imbau.JPG		JPG F...	3 MB	21.11...	11.
03-delhi-eis.JPG	03-delhi-eis.JPG		JPG F...	2 MB	21.11...	11.
04-delhi-ventilator.JPG	04-delhi-ventilator.JPG		JPG F...	3 MB	21.11...	11.
05-delhi-northern-palace-fe...	05-delhi-northern-palace-fens...		JPG F...	3 MB	21.11...	11.
06-delhi-northern-palace-ve...	06-delhi-northern-palace-versi...		JPG F...	3 MB	21.11...	11.
07-delhi-humuyans-tomb.JPG	07-delhi-humuyans-tomb.JPG		JPG F...	2 MB	21.11...	12.
08-delhi-eichhoemchen.JPG	08-delhi-eichhoemchen.JPG		JPG F...	2 MB	21.11...	12.

RegEx (1) ☒ R Match 0 Replace 00 ☐ Include Ext. ☐ Match Case

Repl. (3) ☒ R Replace With ☐ Match Case

File (2) ☒ R Name Keep ☐ Match Case

Case (4) ☒ R Same ☐ Excep.

Remove (5) ☒ R First n 0 Last n 0 From 0 to 0 Chars Words Crop Before ☐ Digits ☐ High ☐ Trim ☐ D/S ☐ Accents ☐ Chars ☐ Sym. Lead Dots Non

Add (7) ☒ R Prefix Insert at pos. 0 Suffix ☐ Word Space

Auto Date (8) ☒ R Mode None Type Creation (Cur) Fmt DMY Sep. Seg. Custom ☐ Cent. Off. 0

Numbering (10) ☒ R Mode None at 0 Start 1 Incr. 1 Pad 0 Sep. Break 0 ☐ Folder Type Base 10 (Decimal) Roman Numerals None

Move/Copy (6) ☒ R None 1 None 1 Sep.

Append Folder Name (9) ☒ R Name None Sep. Levels 1

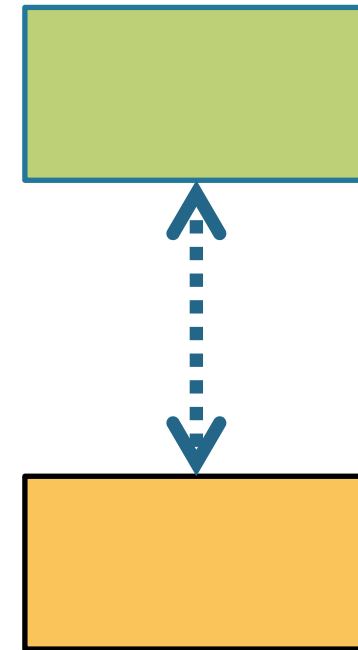
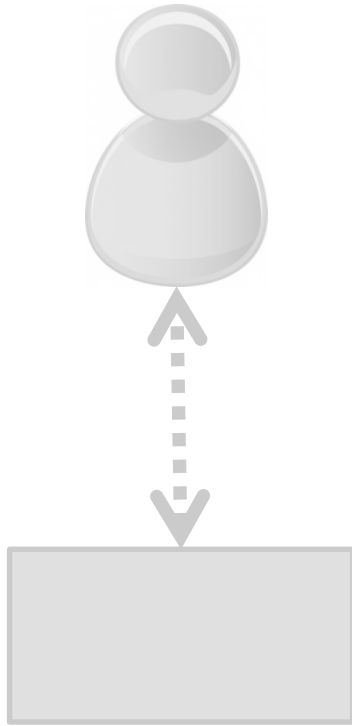
Extension (11) ☒ R Same

Selections (12) Filter * ☒ Folders ☐ Hidden Name Len Min 0 Max 0 ☐ Match Case ☒ Files ☐ Subfolders Path Len Min 0 Max 0

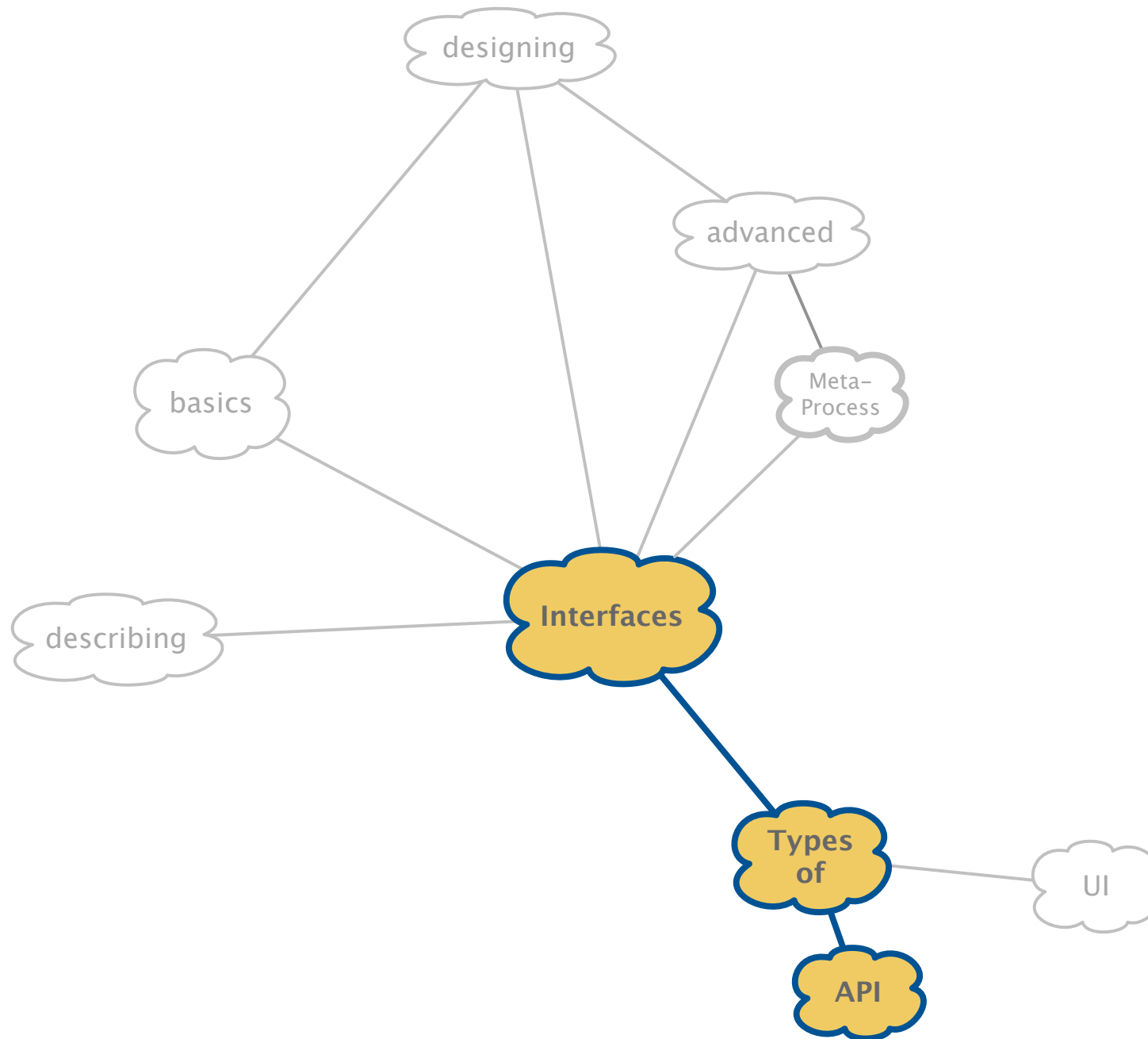
New Location (13) ☐ Copy not Move Path ... Reset Revert Rename

*** Working on multiple computers? Synchronize your files across computers with **ViceVersa PRO**. [Click Here To Find Out More ...](#)

187 Objects (0 Selected) Favourite D:\bilder\2010-Max-Indien\dia-abend.blog



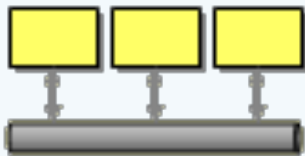
Software Interfaces



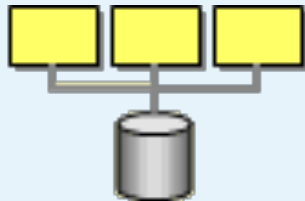
basic types....



Request-Response:
classical with many variants



(async) Messaging



Shared database

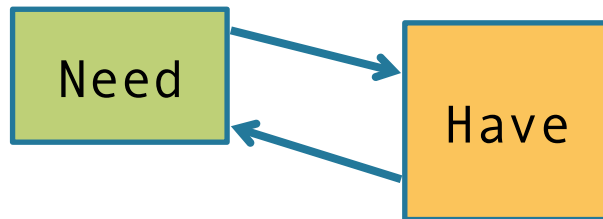


File Transfer

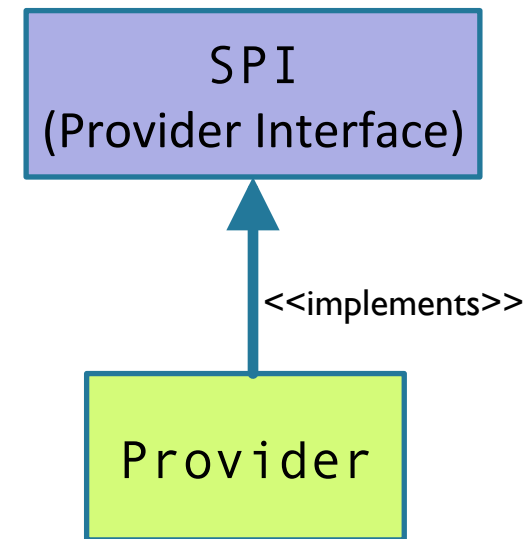


Request-Response...

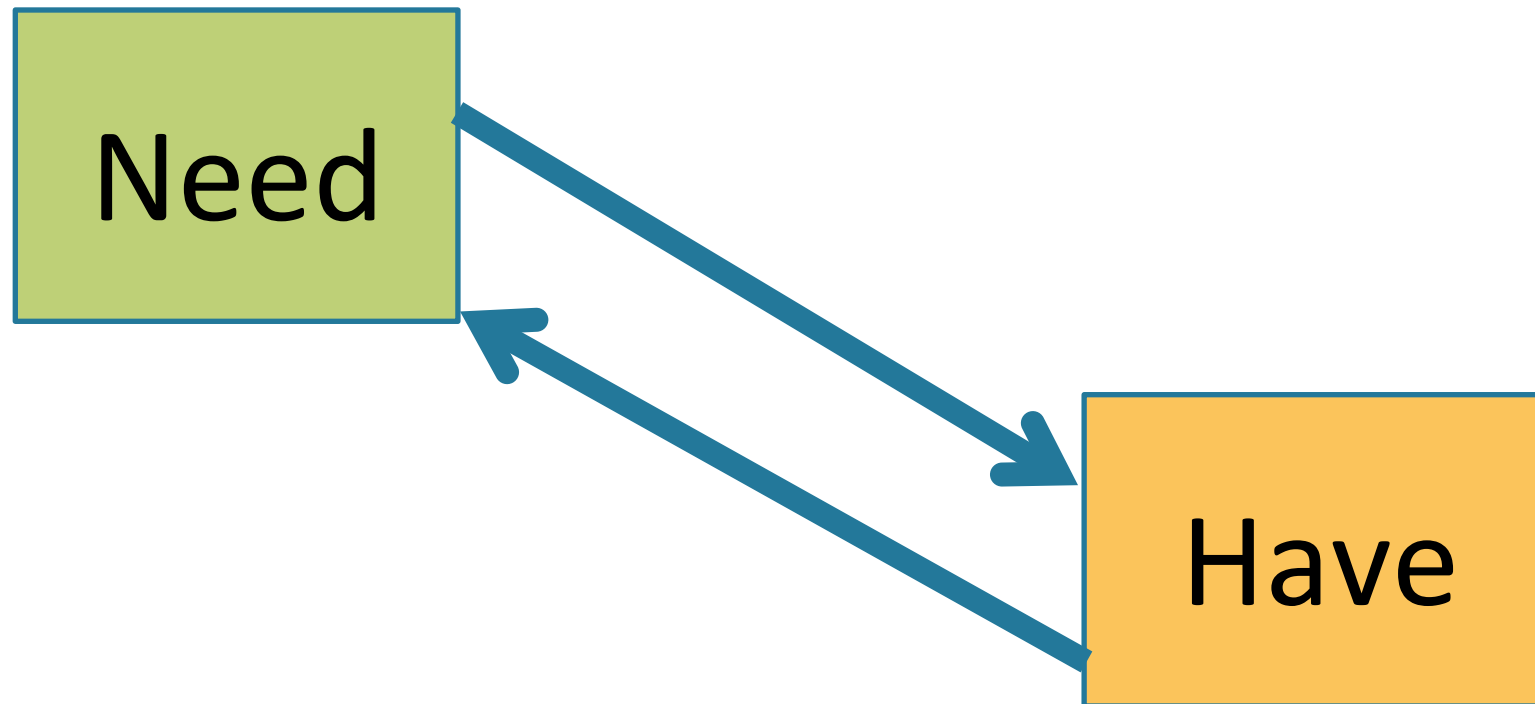
API: use

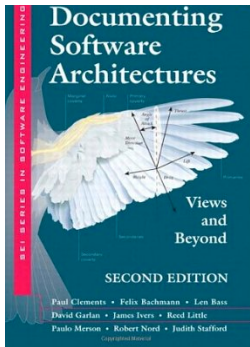


SPI: implement



Intro (V1, API)



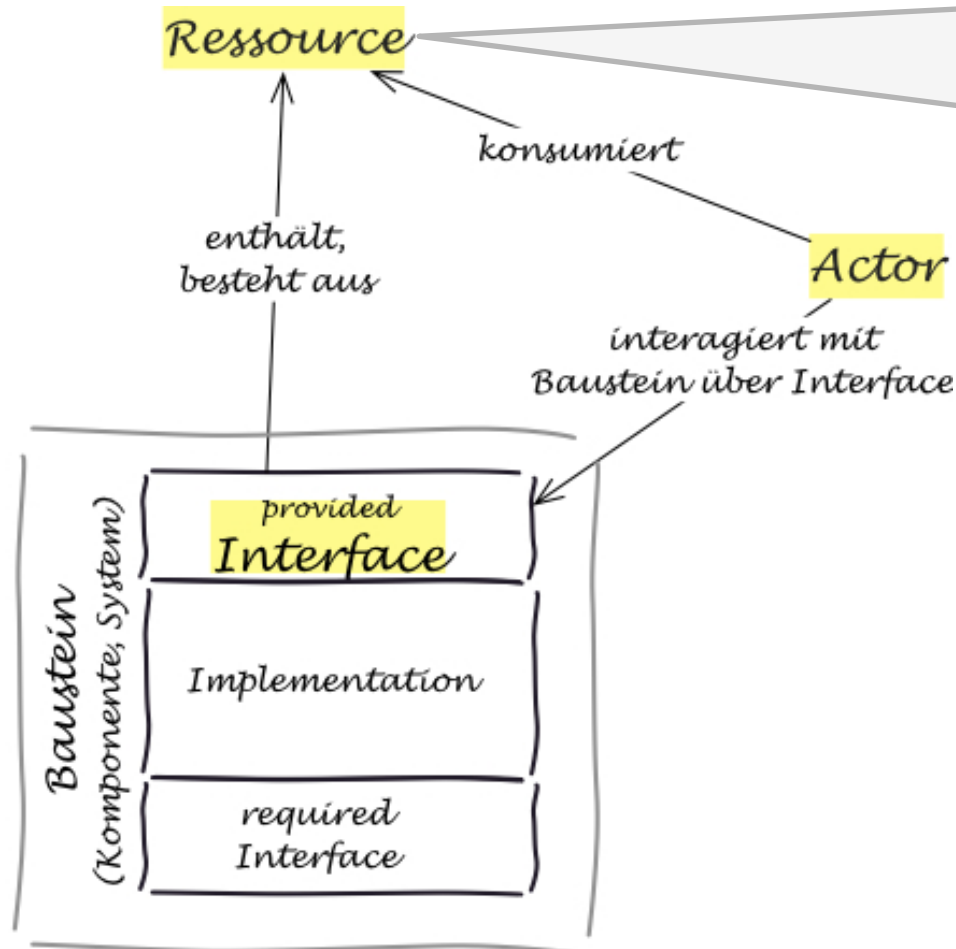


Interface „Metamodel“

== „point of interaction“:

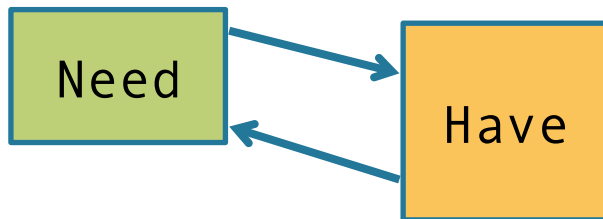
- Function, Method, (Web-)Service
- Data- or Event-Stream
- Shared-Memory

Begriffe aus: Documenting Software Architectures,
Chap. 7: Documenting Software Interfaces



Confusion-warning:
REST-Resources

intro



1. *Need* requires something
(data, service).

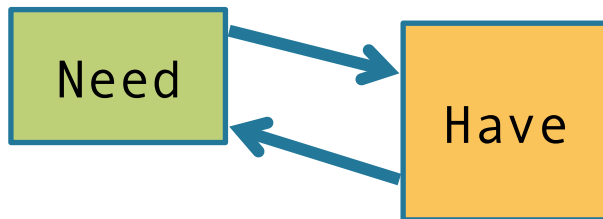
2. Somebody brokers

- determine provider
- establish physical connection
- transmit Need-request to Have

3. Have provides something

**4. Somebody transmits
result
to Need.**

how it works...

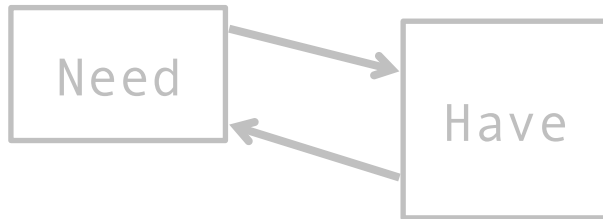


Exchange of resources
(data, service)

between:

- ▶ client, consumer, caller, requester, subscriber, user
- ▶ server, supplier, host, (service-)provider, publisher

you need to care about...



What to transmit?

How does it look like

- Resources
- Methods/functions

What does it mean?

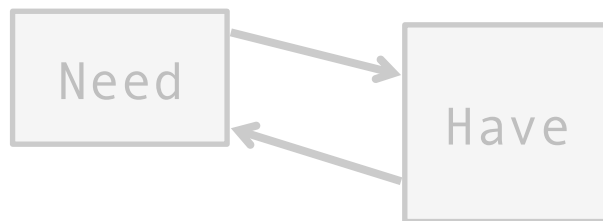
**At
least!**

How Need finds Have

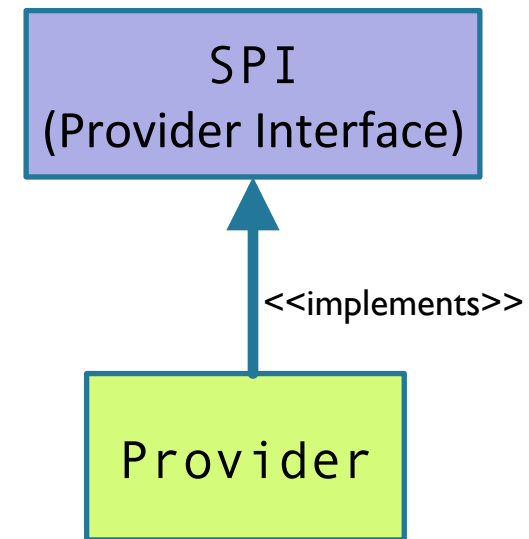


Request-Response...

API: use



SPI: implement

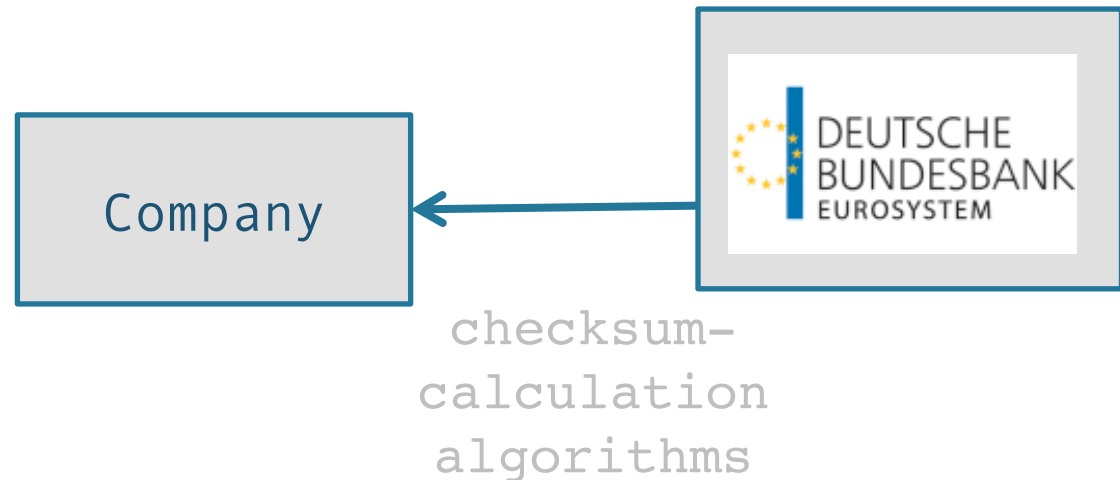


SPI example: Account-Number-Validation

Kennzeichen	Berechnungsmethode für Prüzziffern
00	Modulus 10, Gewichtung 2, 1, 2, 1, 2, 1, 2, 1, 2 Die Stellen der Kontonummer sind von rechts nach links mit den Ziffern 2, 1, 2, 1, 2 usw. zu multiplizieren. Die jeweiligen Produkte werden addiert, nachdem jeweils aus den zweistelligen Produkten die Quersumme gebildet wurde (z. B. Produkt 16 = Quersumme 7). Nach der Addition bleiben außer der Einerstelle alle anderen Stellen unberücksichtigt. Die Einerstelle wird von dem Wert 10 subtrahiert. Das Ergebnis ist die Prüzziffer (10. Stelle der Kontonummer). Ergibt sich nach der Subtraktion der Rest 10, ist die Prüzziffer 0. Testkontonummern: 9290701, 539290858 1501824, 1501832
01	Modulus 10, Gewichtung 3, 7, 1, 3, 7, 1, 3, 7, 1 Die Stellen der Kontonummer sind von rechts nach links mit den Ziffern 3, 7, 1, 3, 7, 1 usw. zu multiplizieren. Die jeweiligen Produkte werden addiert. Nach der Addition bleiben außer der Einerstelle alle anderen Stellen unberücksichtigt. Die Einerstelle wird von dem Wert 10 subtrahiert. Das Ergebnis ist die Prüzziffer (10. Stelle der Kontonummer). Ergibt sich nach der Subtraktion der Rest 10, ist die Prüzziffer 0.
02	Modulus 11, Gewichtung 2, 3, 4, 5, 6, 7, 8, 9, 2 Die Stellen der Kontonummer sind von rechts nach links mit den Ziffern 2, 3, 4, 5, 6, 7, 8, 9, 2 zu multiplizieren. Die jeweiligen Produkte werden addiert. Die Summe ist durch 11 zu dividieren. Der verbleibende Rest wird vom Divisor (11) subtrahiert. Das Ergebnis ist die Prüzziffer. Verbleibt nach der Division durch 11 kein Rest, ist die Prüzziffer 0. Ergibt sich als Rest 1, ist die Prüzziffer zweistellig und kann nicht verwendet werden. Die Kontonummer ist dann nicht verwendbar.

E0 (0 = Null) gültig seit 04.03.2013, Klarstellung zum 03.06.2013	Modulus 10, Gewichtung 2, 1, 2, 1, 2, 1, 2, 1, 2 Die Stellen der Kontonummer sind von rechts nach links mit den Ziffern 2, 1, 2, 1, 2 usw. zu multiplizieren. Die jeweiligen Produkte werden addiert, nachdem jeweils aus den zweistelligen Produkten die Quersumme gebildet wurde (z. B. Produkt 18 = Quersumme 9) plus den Wert 7. Nach der Addition bleiben außer der Einerstelle alle anderen Stellen unberücksichtigt. Die Einerstelle wird von dem Wert 10 subtrahiert. Das Ergebnis ist die Prüzziffer (10. Stelle der Kontonummer). Ergibt sich nach der Subtraktion der Rest 10, ist die Prüzziffer 0. Beispiel: <table><tr><td>Stelle-Nr.</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>Kontonummer</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>8</td><td>0</td><td>1</td><td>3</td></tr><tr><td>Gewichtung</td><td>2</td><td>1</td><td>2</td><td>1</td><td>2</td><td>1</td><td>2</td><td>1</td><td>2</td><td>P</td></tr><tr><td>Produkt</td><td>2</td><td>2</td><td>6</td><td>4</td><td>10</td><td>6</td><td>16</td><td>0</td><td>2</td><td></td></tr><tr><td>Quersumme</td><td>2</td><td>2</td><td>6</td><td>4</td><td>1</td><td>6</td><td>7</td><td>0</td><td>2</td><td></td></tr></table> Summe = 30 + 7 = 37 10 – 7 (Einerstelle) = 3 = Prüzziffer Testkontonummern (richtig):1234568013, 1534568010, 2610015, 8741013011 Testkontonummern (falsch):1234769013, 2710014, 9741015011	Stelle-Nr.	1	2	3	4	5	6	7	8	9	10	Kontonummer	1	2	3	4	5	6	8	0	1	3	Gewichtung	2	1	2	1	2	1	2	1	2	P	Produkt	2	2	6	4	10	6	16	0	2		Quersumme	2	2	6	4	1	6	7	0	2	
Stelle-Nr.	1	2	3	4	5	6	7	8	9	10																																														
Kontonummer	1	2	3	4	5	6	8	0	1	3																																														
Gewichtung	2	1	2	1	2	1	2	1	2	P																																														
Produkt	2	2	6	4	10	6	16	0	2																																															
Quersumme	2	2	6	4	1	6	7	0	2																																															
Ende																																																								

SPI example: Account-Number-Validation



- ▶ Central Bank publishes **list of valid algorithms** to calculate checksums for account-numbers

- ▶ May 2013: approx. 140 different!

- ▶ Companies implement those (or subset)

http://www.bundesbank.de/Navigation/DE/Kerngeschaeftsfelder/Unbarer_Zahlungsverkehr/Pruefzifferberechnung/pruefzifferberechnung.html

SPI (Provider Interface)...

**My implementation will
work forever!**

Expectation of
implementor!

SPI (Provider Interface)...

care for
compatibility

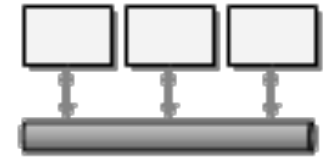
My implementation will
work forever!

SPI (Provider Interface)



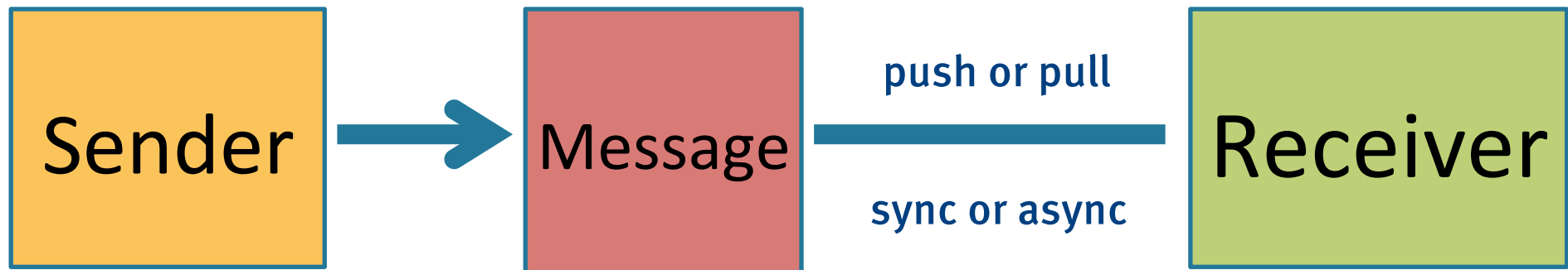
care for
compatibility

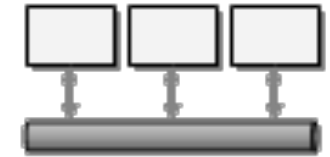
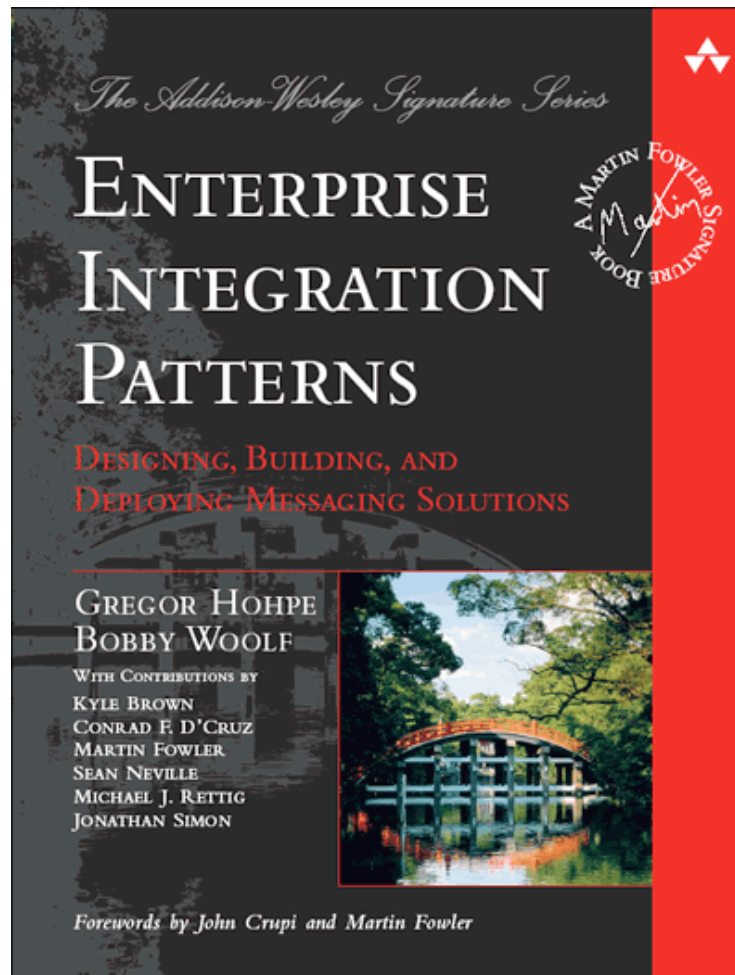
- Don't add methods
(implementors are not compilable)
- Solution: new interfaces...



Intro

(messaging, file-transfer)



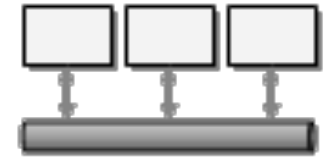


* UGYOC =

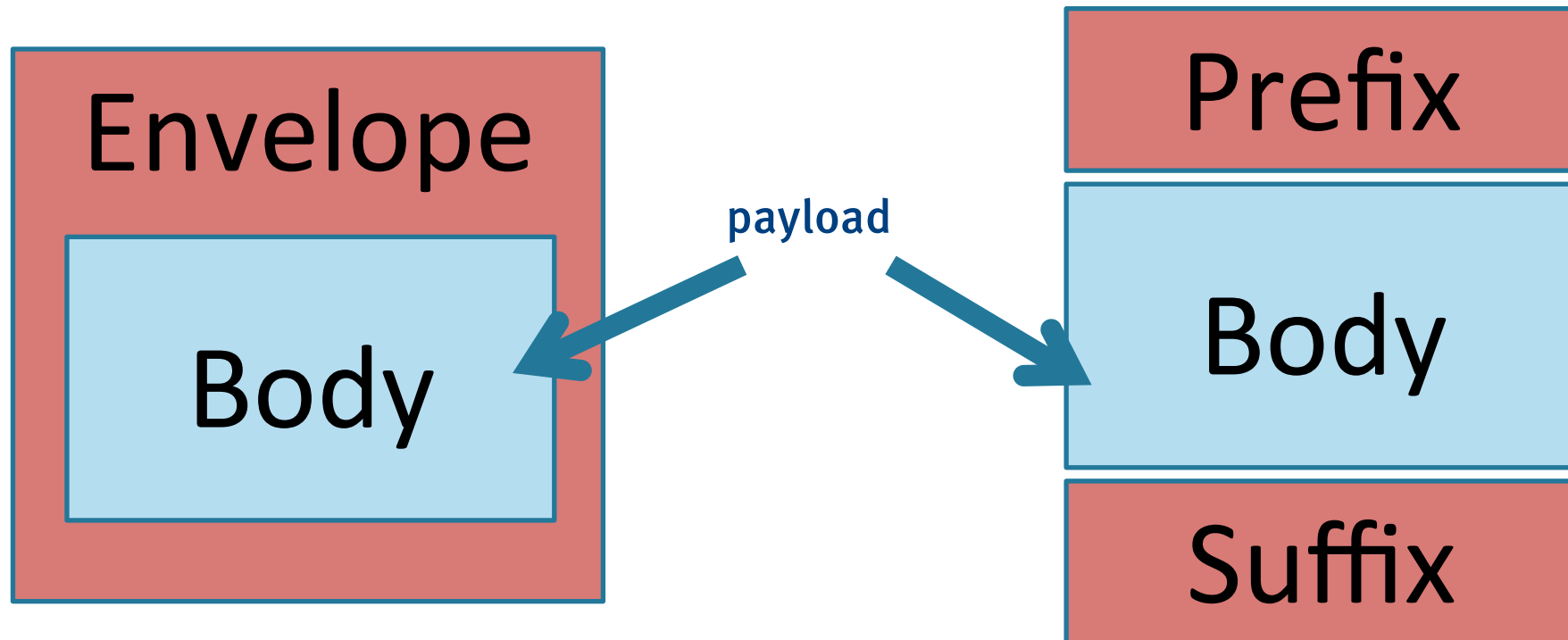
Urgently
Get
Your
Own
Copy

UGYOC*

inno**Q**



Messages, Files



Sample: Chess

Diagram illustrating a Chess engine interface (Anand, Viswanathan) interacting with a Chess engine (Polgar, Judit) via commands.

The interface shows a chessboard, a list of moves, and a command window. The command window displays the following commands:

- command: `Polgar, Judit`
- command: `g7-g5`

The Chess engine (Polgar, Judit) responds with the following moves:

- 18. Qd3-d2
- 19. h5xg6/ep f7xg6
- 20. Bf4xh6
- 21. Qf5xf3
- 22. d4-d5

The interface also displays a table of moves and a command window with the following commands:

- command: `Polgar, Judit`
- command: `g7-g5`

The Chess engine (Polgar, Judit) responds with the following moves:

- 18. Qd3-d2
- 19. h5xg6/ep f7xg6
- 20. Bf4xh6
- 21. Qf5xf3
- 22. d4-d5

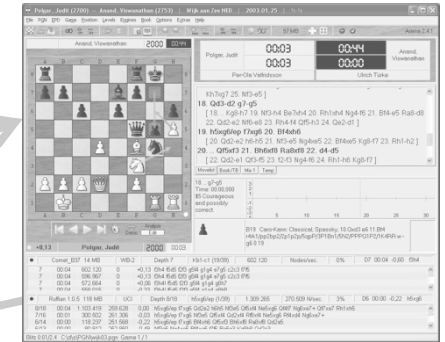
Sample: UCI & xboard

STABLE since
2000 (UCI)!

Chess
engine

command

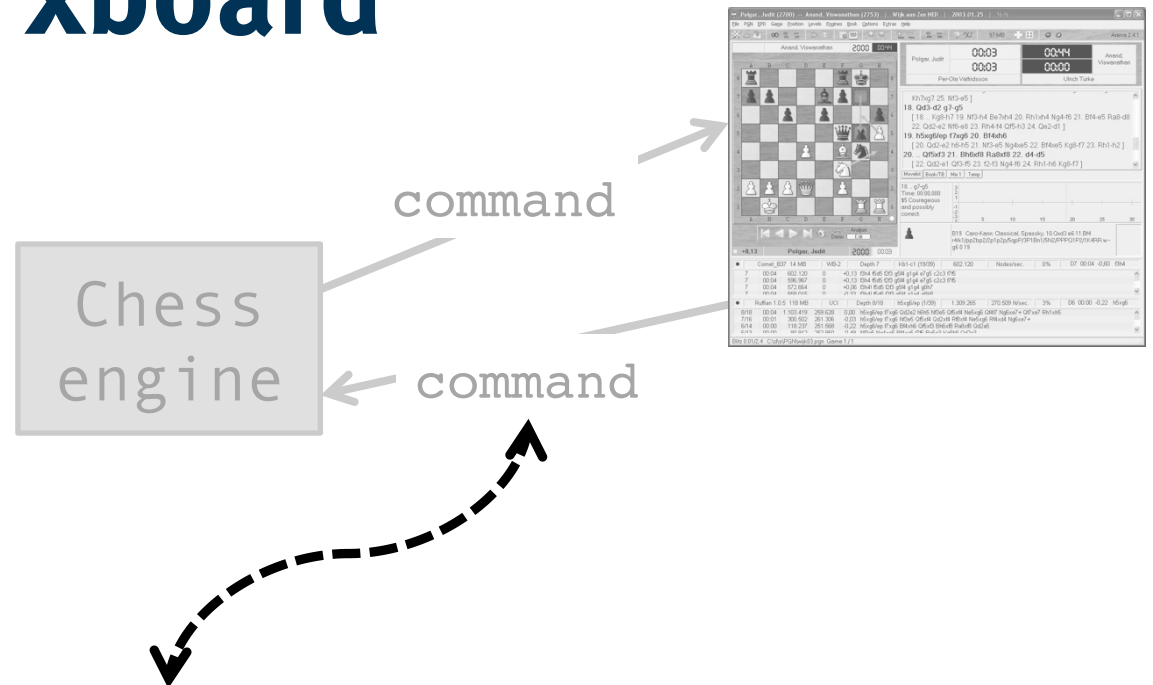
command



„commands“ as stdin/stdout text:

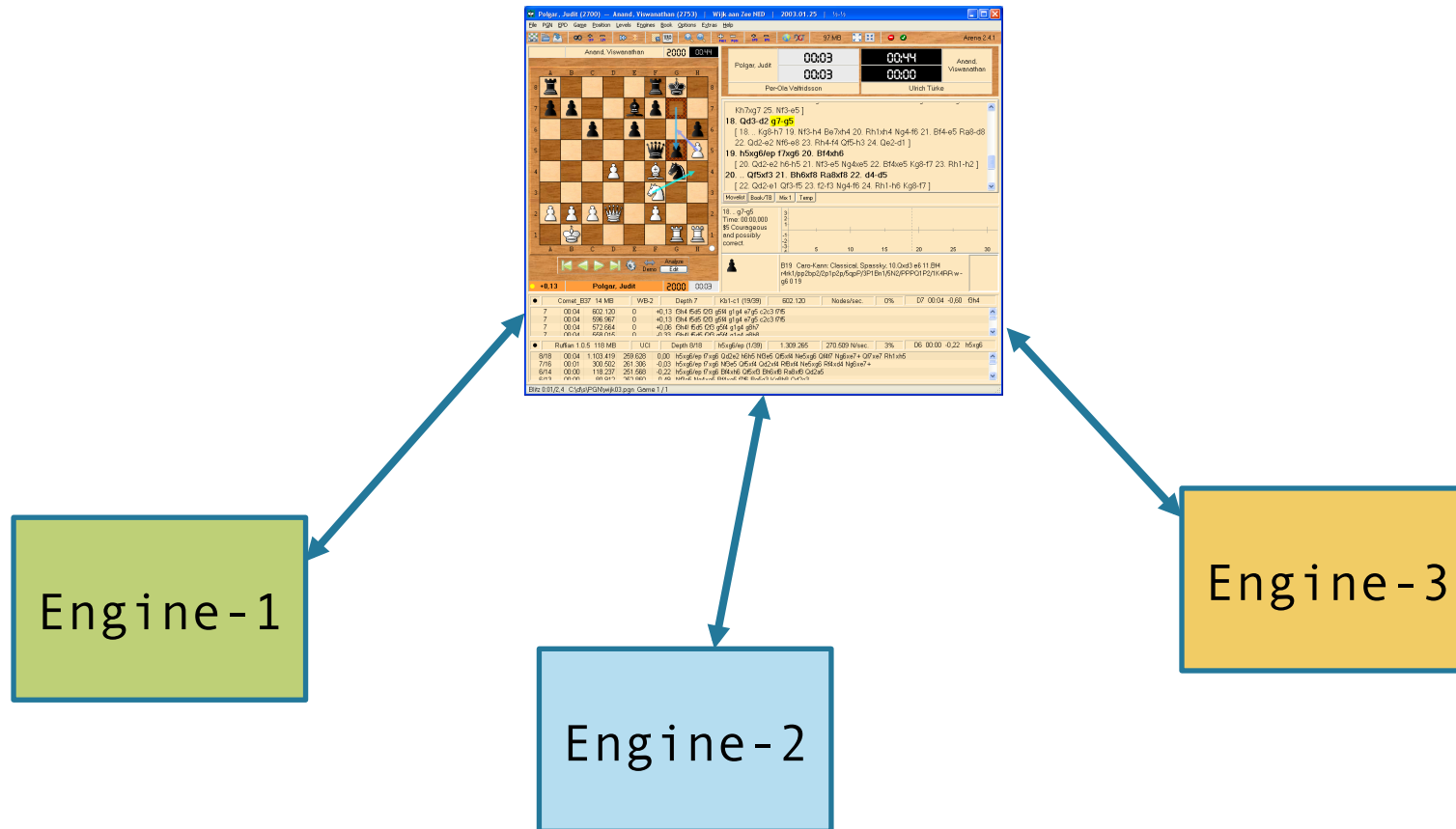
- ▶ ping, setboard, playother, usermove, san, time, draw, sigint, sigterm, myname, variants, colors, memory, smp, option, done ...
- ▶ UCI: <http://www.shredderchess.de/schach-info/features/uci-universal-chess-interface.html>
- ▶ xboard: <http://www.gnu.org/software/xboard/engine-intf.html>

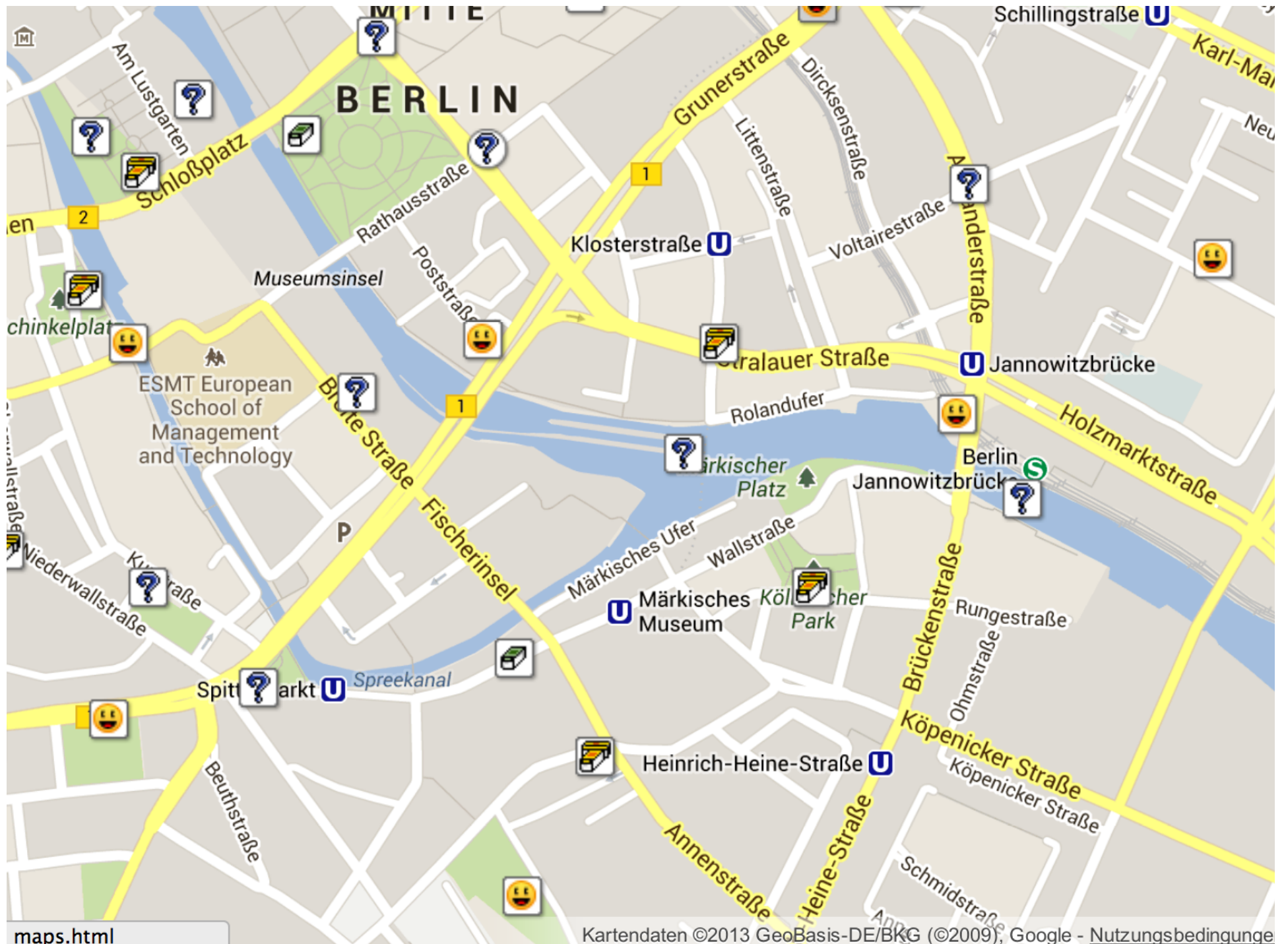
Sample: UCI & xboard



- ▶ Evolution is easy (new commands)
- ▶ Simple, technology neutral
- ▶ But:
 - ▶ Requires parser on both ends

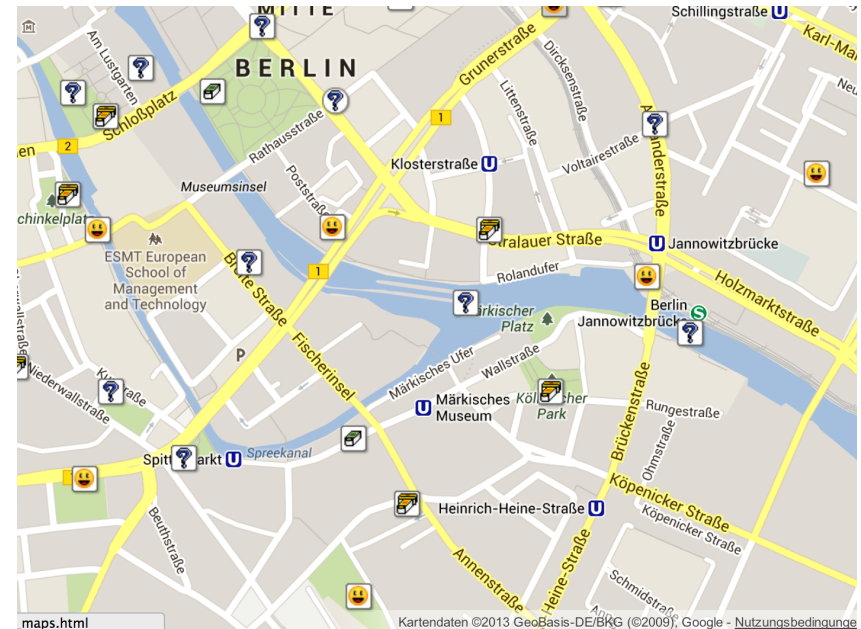
Engine Tournament...





Geocaching...

- ▶ Request Map
- ▶ Select „Cache“
 - ▶ Read description
& other people remarks
- ▶ Find „treasure“
- ▶ Send log-entry to central server



Geocaching...

- Requirements:
 - Works offline
 - Multiple device categories



Sample: Geocaching GPX

Geocaching.com

```
<?xml version="1.0" encoding="utf-8"?>
<gpx xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
  <name>Cache Listing Generated from Geocaching.com</name>
  <desc>This is an individual cache generated from Geocaching.com</desc>
  <author>Account "mgo" From Geocaching.com</author>
  <email>contact@geocaching.com</email>
  <url>http://www.geocaching.com</url>
  <urlname>Geocaching - High Tech Treasure Hunting</urlname>
  <time>2006-02-11T19:34:49.3158398-08:00</time>
  <keywords>cache, geocache</keywords>
  <bounds minlat="48.89581666666667" minlon="9.153916666666667" maxlat="48.89581666666667" maxlon="9.
  <wpt lat="48.89581666666667" lon="9.153916666666667">
    <time>2005-09-28T00:00:00.0000000-07:00</time>
    <name>GCQQK3</name>
    <desc>"Red House" by mgo, Traditional Cache (1.5/2.5)</desc>
    <url>http://www.geocaching.com/seek/cache_details.aspx?guid=89c6e40f-a26b-4fa6-b074-d482af76f
    <urlname>"Red House"</urlname>
    <sym>Geocache</sym>
    <type>Geocache|Traditional Cache</type>
    <groundspeak:cache id="296768" available="True" archived="False" xmlns:groundspeak="http://ww
      <groundspeak:name>"Red House"</groundspeak:name>
      <groundspeak:placed_by>mgo</groundspeak:placed_by>
      <groundspeak:owner id="153659">mgo</groundspeak:owner>
      <groundspeak:type>Traditional Cache</groundspeak:type>
      <groundspeak:container>Regular</groundspeak:container>
      <groundspeak:difficulty>1.5</groundspeak:difficulty>
      <groundspeak:terrain>2.5</groundspeak:terrain>
      <groundspeak:country>Germany</groundspeak:country>
      <groundspeak:state>
      </groundspeak:state>
      <groundspeak:short_description html="False">
      </groundspeak:short_description>
```



Interfacing with Files



- ▶ Still widely used in practice, e.g. banking, insurance, government
- ▶ Technology-neutral
- ▶ Flexible
- ▶ Easy to use & test

Prefix

Body

Suffix



Interfacing with Files



- ▶ Requires **specific** parser and generator
- ▶ REST-free
- ▶ Un-web, un-mobile
- ▶ Boring, old-fashioned

Prefix

Body

Suffix



Interfacing with Files

- ▶ Prefix / Suffix can contain:
 - ▶ Sender, Receiver info
 - ▶ Processing instructions
 - ▶ Accounting info
 - ▶ Payload description
 - ▶ Other „meta“ stuff





Interfacing with Files

https://www.ksk-koeln.de/formatbeschreibung_datentraegeraustausch.pdf

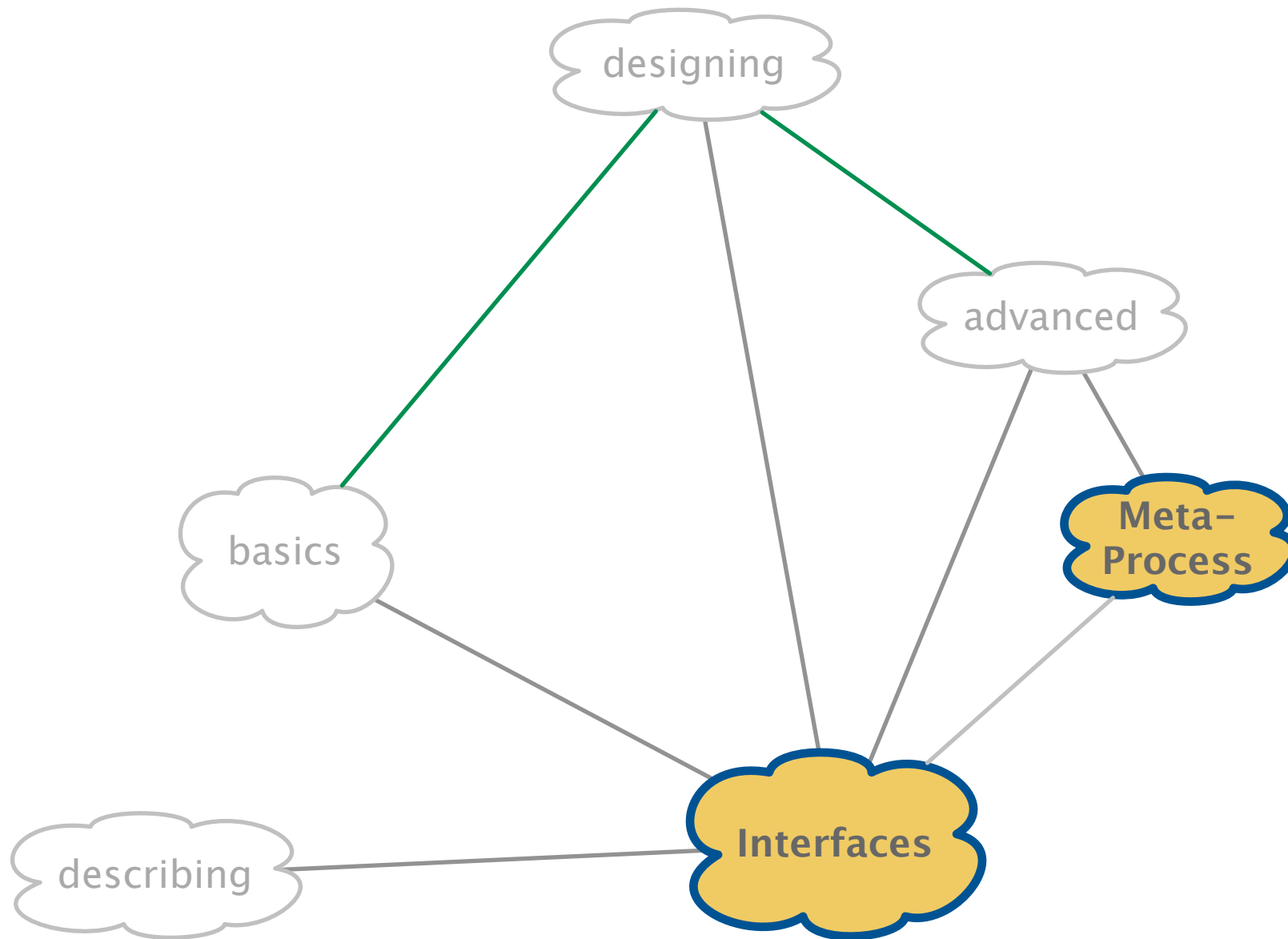
Kreissparkasse Köln

- Formatbeschreibung DTAUS -

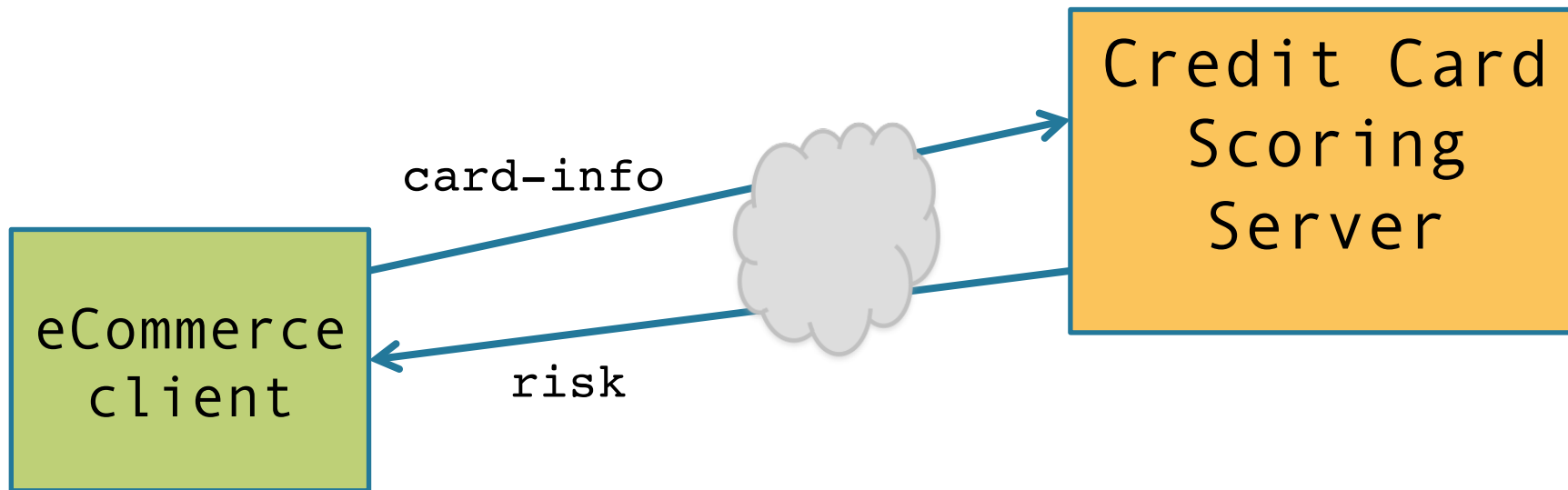
Datensatz A: Datenträger-Vorsatz

Nr.	Länge	Typ	Inhalt	Erläuterung
1	4	n	Satzlänge	'0128'
2	1	a	Satzart	Konstante "A"
3	2	a	Kennzeichen "GK" bzw. "LK" "GB" bzw. "LB"	Hinweis auf Gutschriften (G) bzw. Lastschriften (L), Kundendiskette (K), Bankdiskette (B)
4	8	n	Bankleitzahl	Bankleitzahl des Kreditinstituts (Diskettenempfänger)
5	8	n	X'F0'	-
6	27	a	Auftraggeber	Auftraggeber bzw. Diskettenabsender
7	6	n	Datum	Datei- bzw. Diskettenerstellungsdatum im Format TTMMJJ
8	4	a	Leerzeichen X'20'	-
9	10	n	Kontonummer	Begünstigter / Auftraggeber Kunde, max. 10 Stellen Über dieses Konto wird der Gegenwert verrechnet
10	10	n	Referenznummer des Einreichers	Referenznummer des Einreichers
11a	15	a	Leerzeichen	Reserve

Prefix



Sample: Credit Card Scoring (1)



Sample:

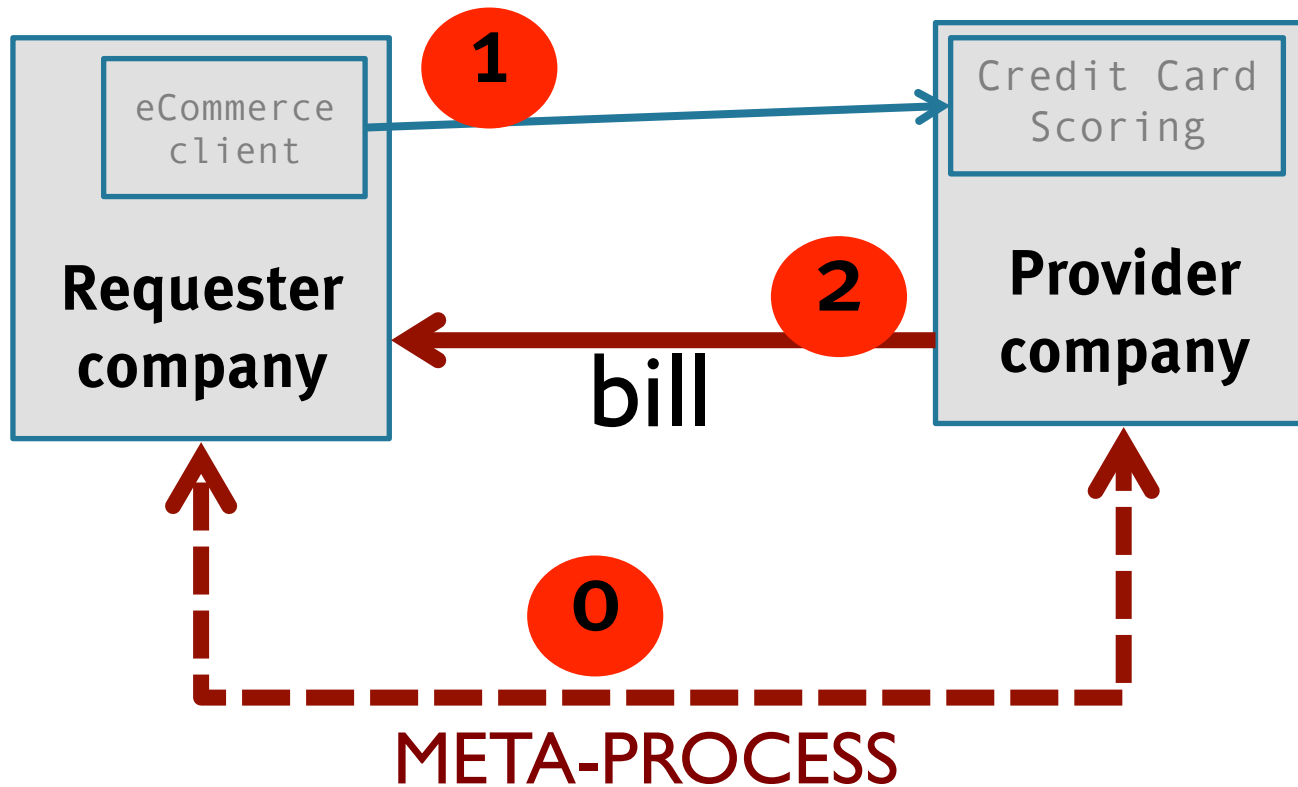
Credit Card Scoring (2)



Function:	getBasicScore()		
Description:	Evaluates risks for specific credit card (score). Errors are returned as ArrayOfError.		
Return:	ScoringResult		
Parameter:	Type	Name	Remark
1	Auth	Auth	Authencation-Parameter of Client
2	Suspect	Suspect	Card and holder
3	String	Reason	reason
Optional 4	[String]	Params	Additional parameters

From: http://www.mediafinanz.de/upload/soap/SOAP_Infoscore.pdf

(heavy) violence...





Meta-Process...

- ▶ Human activities before and after software interaction
- ▶ Consequences:
 - ▶ Financial
 - ▶ Legal
 - ▶ Organizational





Meta-Process...

- ▶ Verification of usage:
Legally valid logging
- ▶ Authorization
- ▶ Tariff, pricing, contract

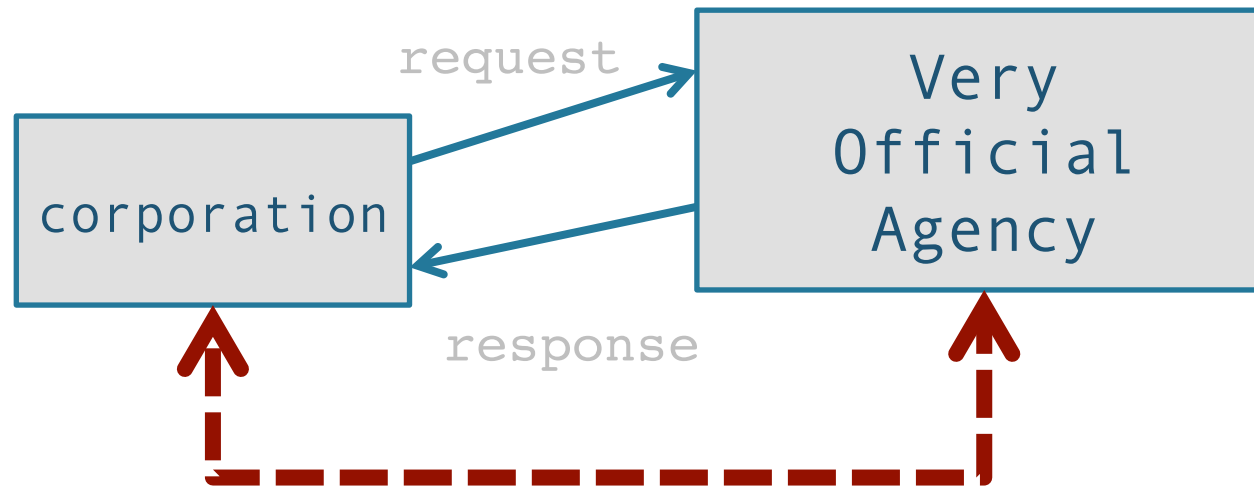


Meta-Process...

- ▶ **Guarantees:**
Quality of Service
- ▶ Versioning
- ▶ **Stakeholder
inferno...**
- ▶ <LOUS*>

* LOUS: Lot's Of Ugly Stuff

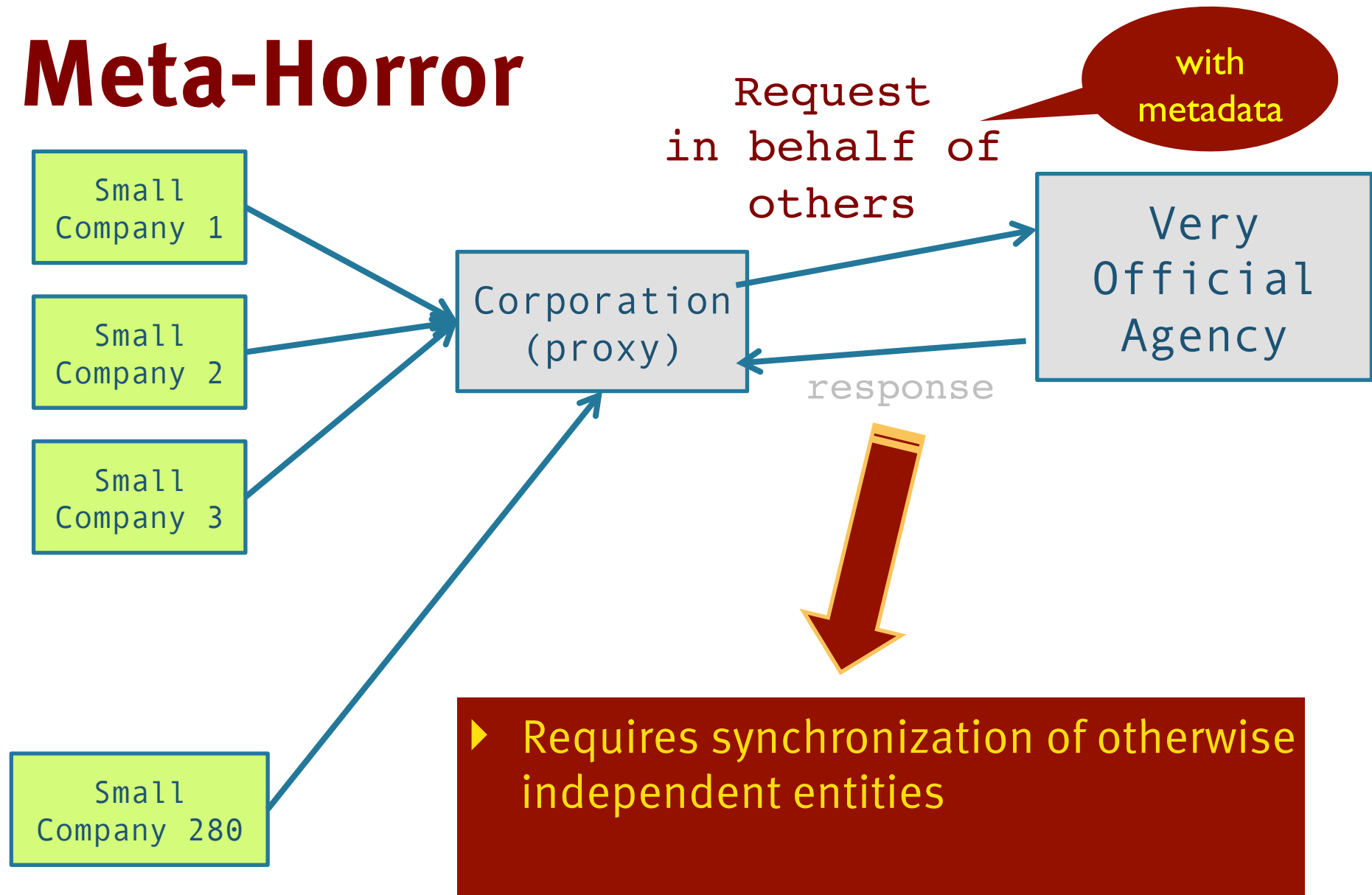
Another Example...



- ▶ „request“ contains meta info:
requestCount: „How often did you call me?“
 - ▶ E.g.: pension contributions

- 
- ▶ Requires metadata management...
 - ▶ Calls CAN NOT be independent

Meta-Horror



4 Meta-Process requires...

governance

governance

management

management

API Management is relevant...

FORRESTER®

The Forrester Wave™: API Management Platforms, Q1 2013

by Eve Maler and Jeffrey S. Hammond, February 5, 2013

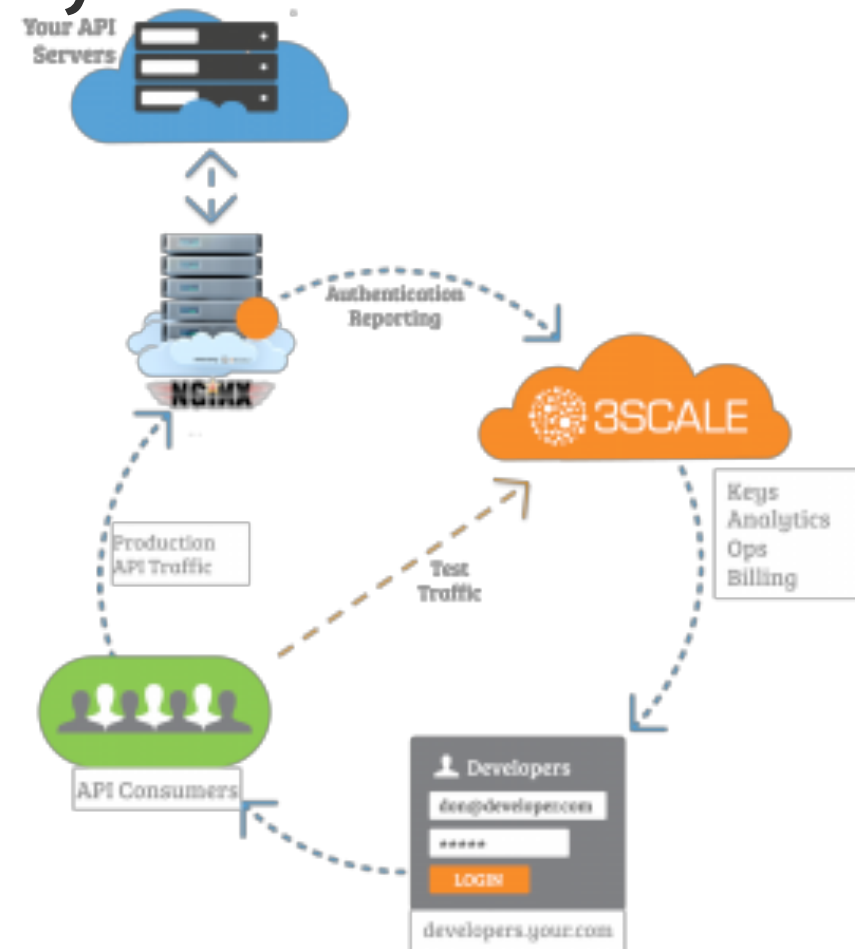
API Management is Distinct From SOA Governance

API Management Platforms Will Continue To Develop

innoQ

API Management ?

- ▶ Access Control & Security
- ▶ Developer Portal
- ▶ Contracts, Rates
- ▶ Analytics
- ▶ Billing, Payment
- ▶ Deployment
- ▶ Support, SLA



<http://www.3scale.net/api-management/api-proxy-for-api-traffic-management-by-3scale/>

innoQ

API Management is BIG...



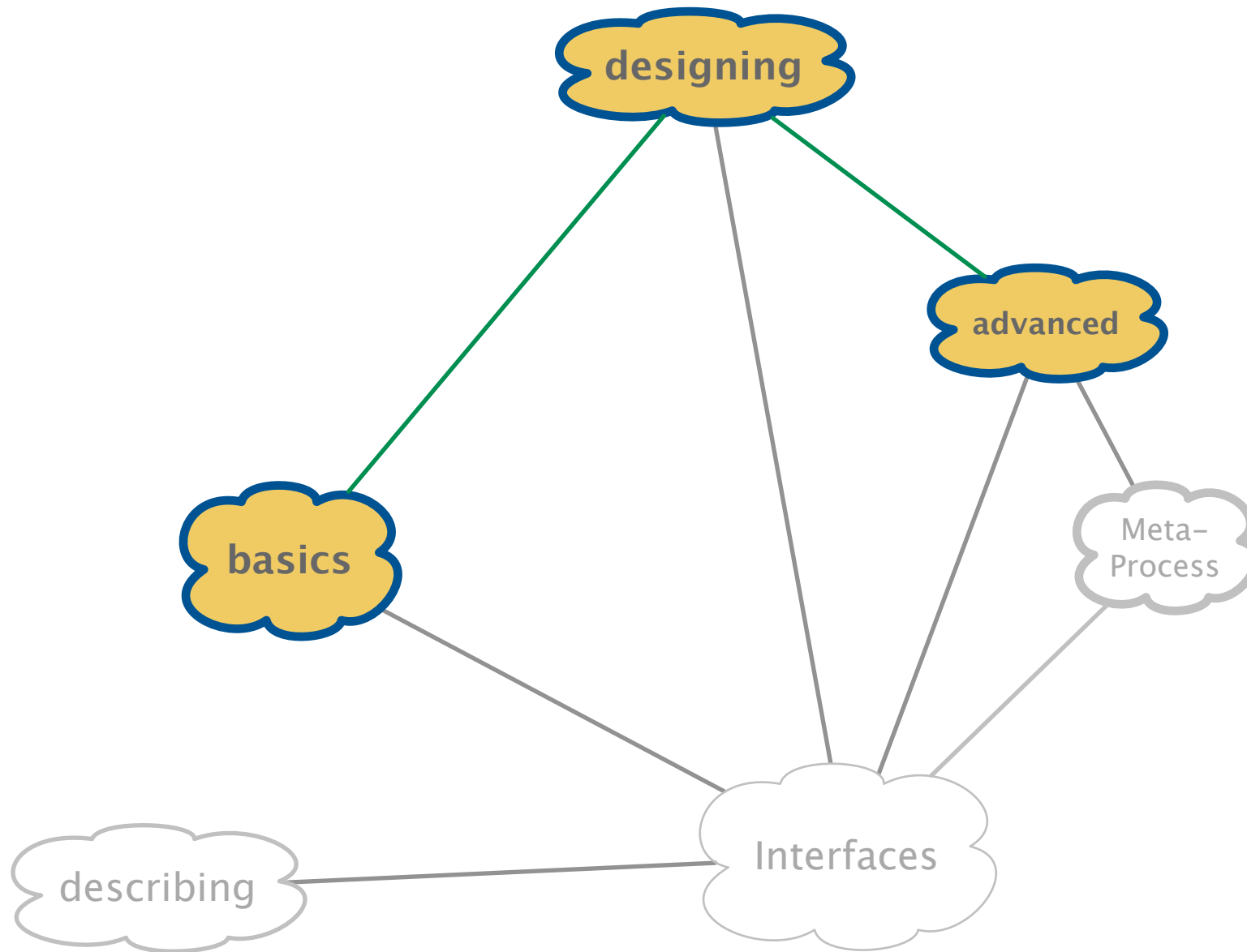
SOA | software™

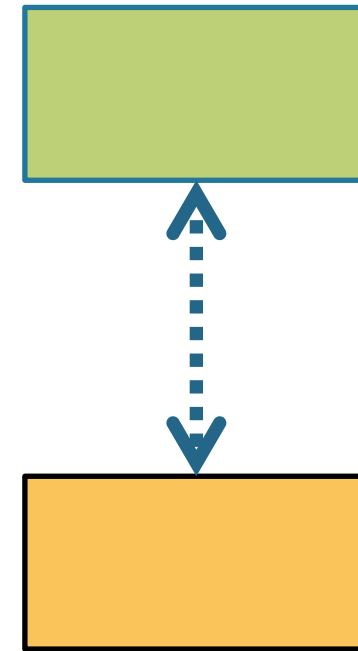


WEBSERVIUS

<http://blog.programmableweb.com/2011/10/19/api-service-provider-roundup/>







Designing Software Interfaces

What makes a good API?

Comfortable
for
API users

- ▶ Easy to learn and use
- ▶ Leads to readable code
- ▶ Complete
- ▶ Hard to misuse
- ▶ Easy to extend

Josh Bloch / Jasmin Blanchette

API == Matter of Taste

```
import com.beust.jcommander.Parameter;

public class JCommanderExample {
    @Parameter
    private List<String> parameters = new ArrayList<String>();

    @Parameter(names = { "-log", "-verbose" }, description = "Level of verbosity")
    private Integer verbose = 1;

    @Parameter(names = "-groups", description = "Comma-separated list of group names to be run")
    private String groups;

    @Parameter(names = "-debug", description = "Debug mode")
    private boolean debug;
}
```

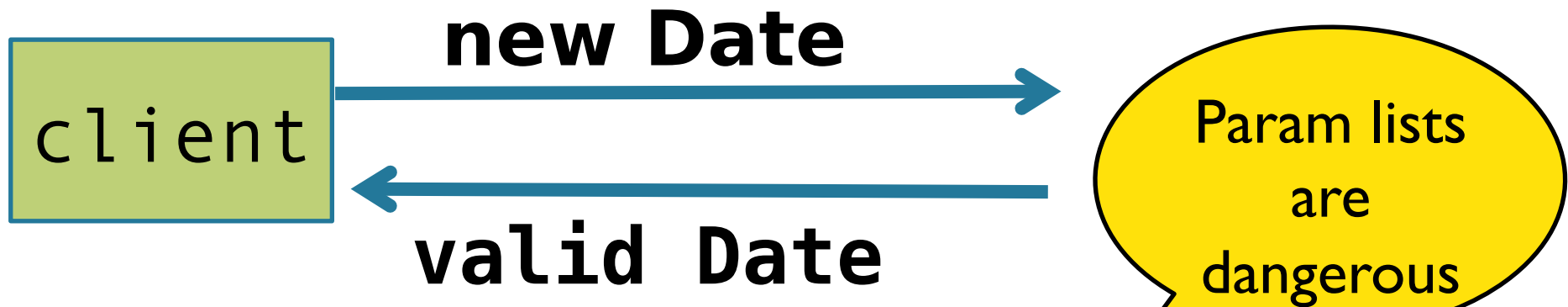
```
/*
 *
 */
parser = new OptionParser() {
    {
        /* required arguments */
        /* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
        acceptsAll(asList("f", "files", "sourcefiles"),
            "non-empty list of source/input files to be processed, separated by ':'")
            .withRequiredArg()
            .required()
            .describedAs("path1:" + ',' + "path2:...")
            .ofType(String.class)
            .withValuesSeparatedBy(':');

        acceptsAll(asList("t", "target", "targetfile", "outputfile", "outfile"),
            "The output pdf file created by concatenation, encryption or metadata-enrichment")
            .withRequiredArg()
            .ofType(String.class)
            .defaultsTo( DEFAULT_TARGET_FILE );

        /* optional arguments */
        /* ===== */
        acceptsAll(asList("h", "help"),
            "display this help message and exit")
            .withNoRequiredArg()
            .required(false)
            .describedAs("display this help message and exit")
            .ofType(String.class)
            .defaultsTo(DEFAULT_HELP_FILE);
    }
}
```

Tipps for API design (1)

- ▶ **Beware** of meta process
- ▶ If in doubt, remove it



new Date(...):

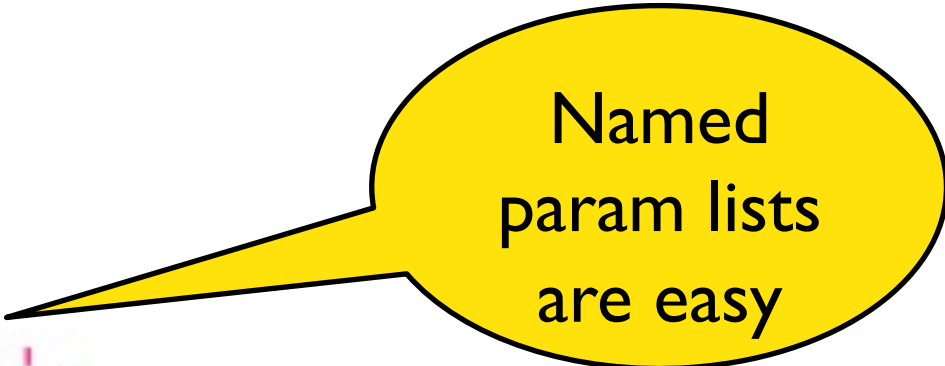
- (1) Date(y, m, d)?**
- (2) Date(d, m, y)?**
- (3) Date("23/5/13")?**

Why don't we all ...

► Use Groovy? Or Python?

```
class Address {  
    String street  
    String city  
    int number  
}
```

```
def a = new Address(  
    street: 'Main St.',  
    number: 1,  
    city: 'Cologne')
```



Named
param lists
are easy

Tipps for API design (2)

- ▶ **Know your clients:**
 - ▶ Perform „stakeholder analysis“
 - ▶ Be sure to understand client use-cases
 - ▶ Create client-type specific interfaces
- ▶ Use it yourself. Again and again.

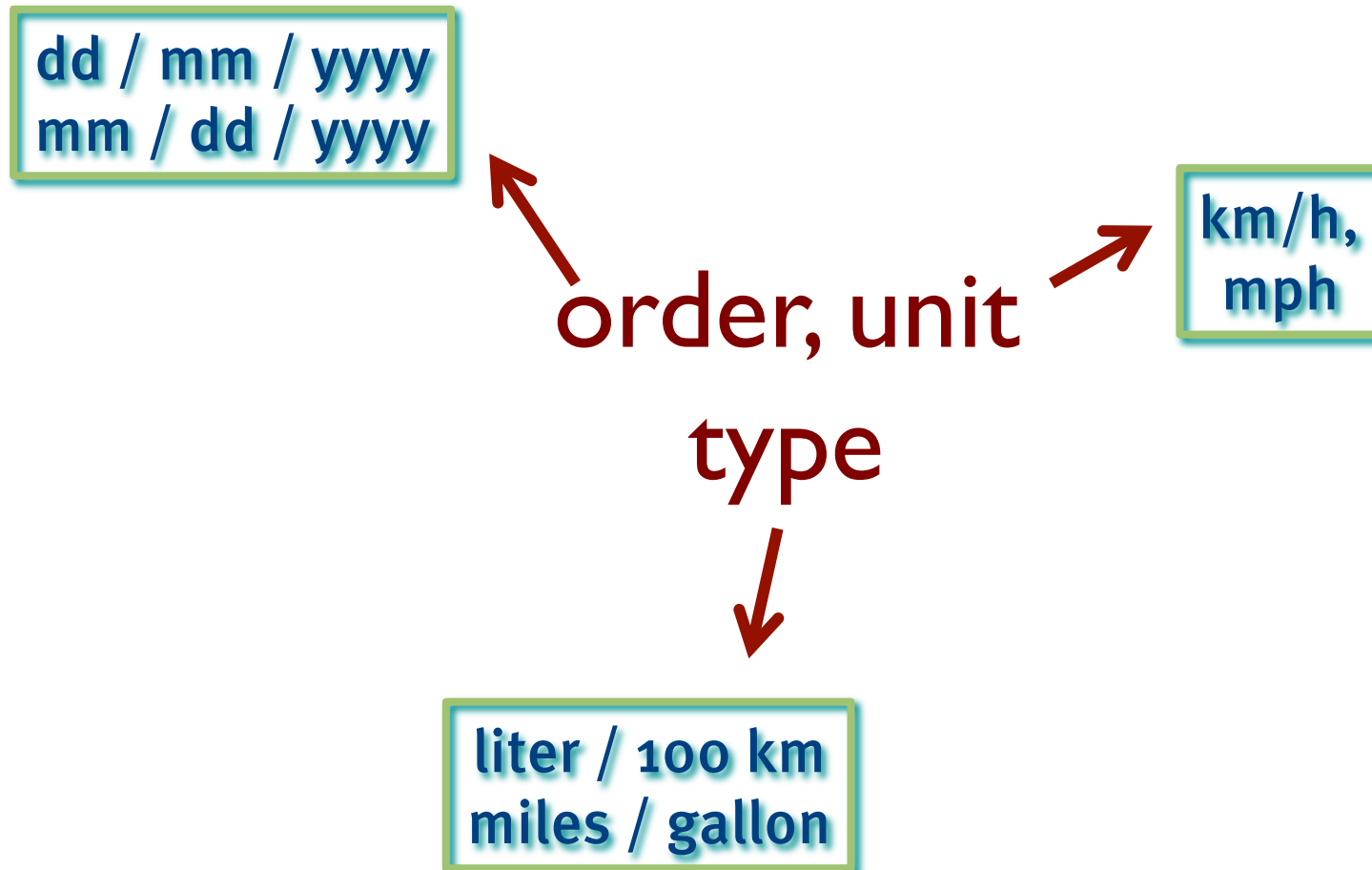
Tipps for API design (3)

- ▶ **Start with examples:**
 - ▶ Write clients, using your interface
 - ▶ Write tests (TDD philosophy)
- ▶ Show it to somebody **ELSE**

Tipps for API design (4)

- ▶ Names are important
 - ▶ API shall be understandable for clients
 - ▶ Josh Bloch calls it „Little Language“
 - ▶ Use client-specific terminology
- ▶ See: Domain-Driven-Design 😊

e.g. habits, fashions, practices



Improve Readability

```
var finder = new PersonFinder();  
finder.FirstName = "Bill";  
finder.LastName = "Gates";  
finder.IncludePhoto = true;  
var persons = finder.Find();
```



```
var persons = new PersonFinder()  
    .FirstName("Bill")  
    .LastName("Gates")  
    .IncludePhoto  
    .Find();
```



in Practice...

JOptSimple, Command Line Parsing Library

```
package joptsimple.examples;

import joptsimple.OptionParser;
import joptsimple.OptionSet;
import org.junit.Test;

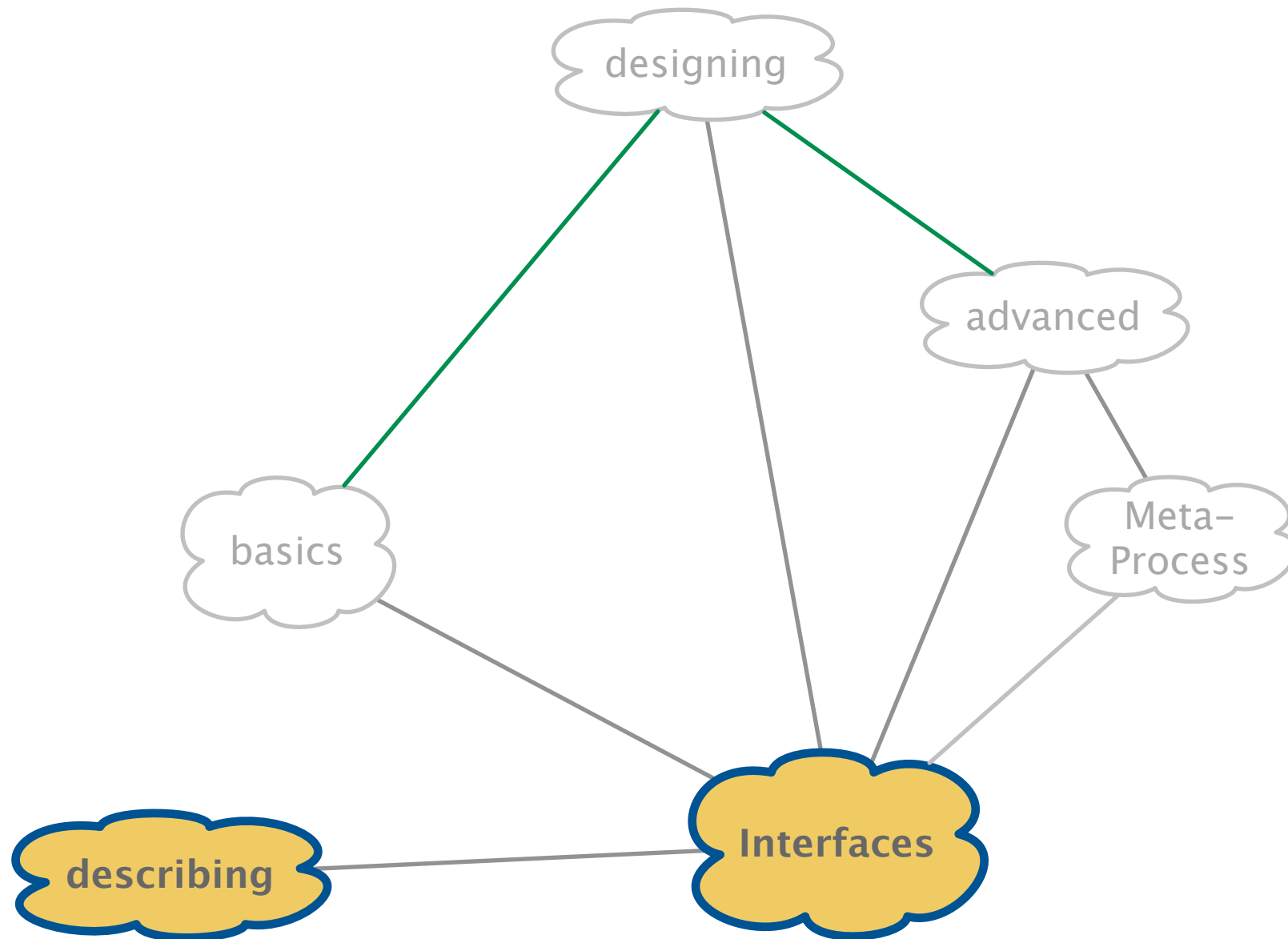
import static org.junit.Assert.*;

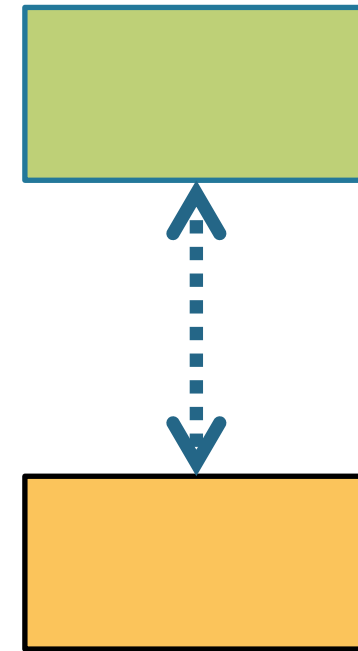
public class OptionArgumentValueTypeTest {
    @Test
    public void convertsArgumentsToJavaValueTypes() {
        OptionParser parser = new OptionParser();
        parser.accepts( "flag" );
        parser.accepts( "count" ).withRequiredArg().ofType( Integer.class );
        parser.accepts( "level" ).withOptionalArg().ofType( Level.class );

        OptionSet options = parser.parse( "--count", "3", "--level", "DEBUG" );

        assertTrue( options.has( "count" ) );
        assertTrue( options.hasArgument( "count" ) );
        assertEquals( Integer.valueOf( 3 ), options.valueOf( "count" ) );

        assertTrue( options.has( "level" ) );
        assertTrue( options.hasArgument( "level" ) );
        assertEquals( Level.DEBUG, options.valueOf( "level" ) );
    }
}
```





Documenting Software Interfaces

Reasons for documenting interfaces

▶ allow others to use your system

Web / REST!

▶ Generate integration code from documentation

▶ show interactions between building blocks

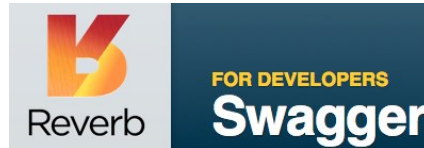
▶ impact analysis

▶ assure long-term understandability

Options for documenting interfaces

1. RESTful (for web stuff)

Very hipp!



I/O DOCS
THE COOLEST, FASTEST WAY

2. Intra-app (your own stuff)

Documenting RESTful interfaces (4)

- Swagger: produce, visualize & consume RESTful services

The screenshot shows the Swagger UI interface for a REST API. At the top, there is a green header bar with the Swagger logo, a search icon, a URL input field containing `http://petstore.swagger.wordnik.com/api/api-docs.json`, a text input field containing `special-key`, and an **Explore** button.

Below the header, the interface displays the **/user** endpoint. It lists three POST operations:

- POST /user.json/createWithArray**: Creates list of users with given input array
- POST /user.json**: Create user
- POST /user.json/createWithList**: Creates list of users with given list input

The **Parameters** section is expanded for the **POST /user.json/createWithList** operation. It shows a table with the following columns: Parameter, Value, Description, and Data Type.

Parameter	Value	Description	Data Type
List[User]	<code>{required}</code> Parameter content type: application/json	List of user object	List[User] class User (id: long, lastName: string, phone: string, username: string, email: string, userStatus: int = ['1-registered' or '2-active' or '3-closed'] {User Status}, firstName: string, password: string)

Below the parameters table, there is a **Try it out!** button.

At the bottom of the interface, there are three more operations listed:

- PUT /user.json/{username}**: Updated user
- DELETE /user.json/{username}**: Delete user
- GET /user.json/{username}**: Get user by user name

Template for Documenting RESTful interfaces

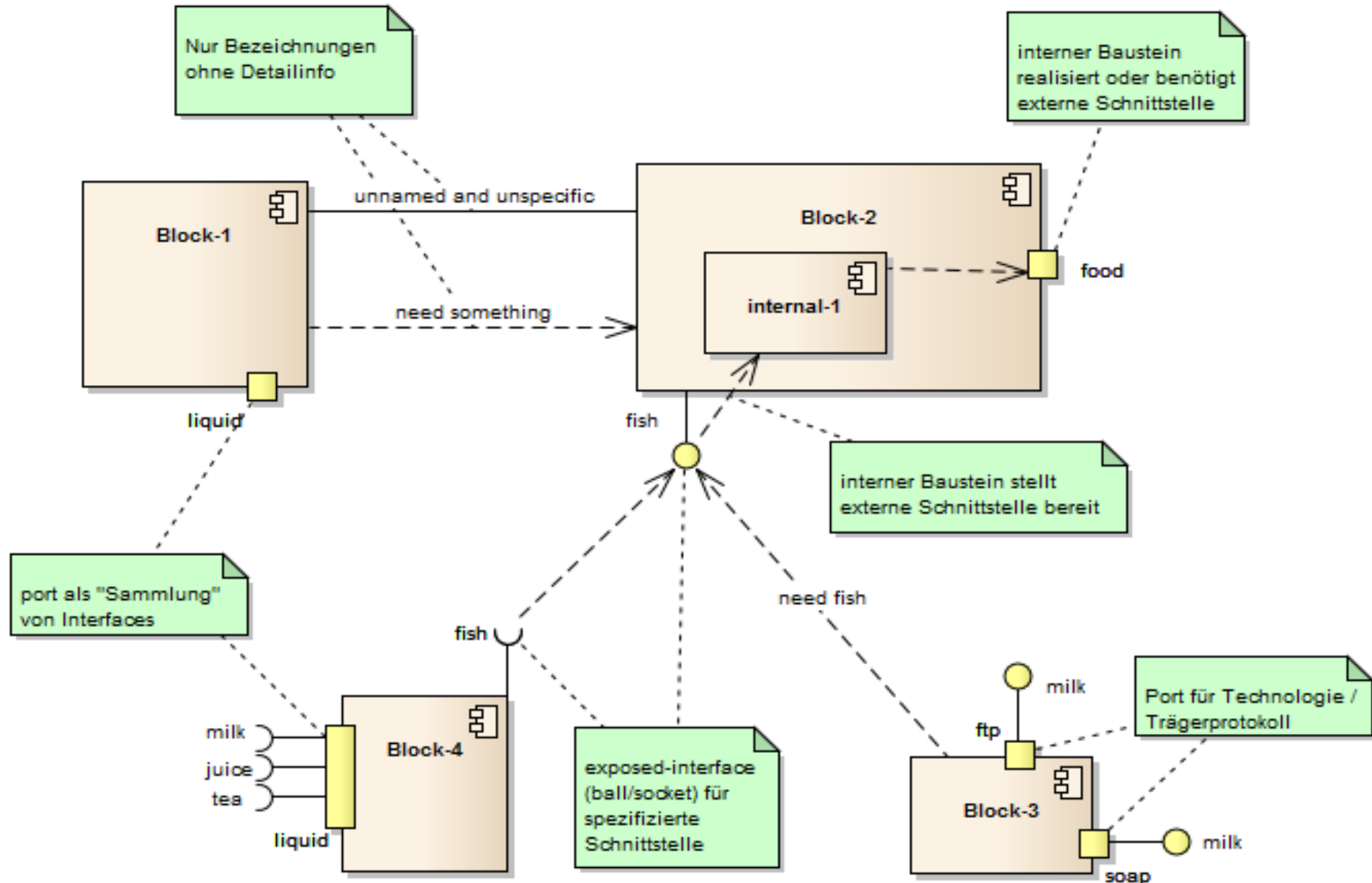
Title	⟨The name of your API call⟩ Example : Show All Users
URL	⟨The URL structure (path only, no root url)⟩ /users or /users/:id or /users?id=:id
Method	The request type⟩ : GET POST DELETE PUT
URL Params	If URL params exist, specify them in accordance with name mentioned in URL section. Separate into optional and required.⟩
Data Params	⟨If making a post request, what should the body payload look like? This is a good time to document your various data constraints too.⟩
Success Response	⟨What should the status code be on success and is there any returned data? This is useful when people need to know what their callbacks should expect!⟩
Error Response	⟨Most endpoints will have many ways they can fail. From unauthorized access, to wrongful parameters etc. All of those should be listed here. It might seem repetitive, but it helps prevent assumptions from being made where they shouldn't be.⟩
Sample call	⟨Just a sample call to your endpoint in a runnable format (\$.ajax call or a curl request) - this makes life easier and more predictable.⟩

UML for documentation

- ▶ Often mis-used
- ▶ offers **too MANY** options
 - ▶ Go for simplicity (selective lazyness)
 - ▶ More precision / details => more effort required
- ▶ Can
 - ▶ help with overview
 - ▶ explain context

Interfaces in UML

cmp Options-for-Interfaces



Template for Interface documentation

Identifikation: Genaue Bezeichnung und Version der Schnittstelle.

Bereitgestellte Ressource(n):

Welche Ressourcen (Daten, Ereignisse) stellt diese Schnittstelle ihren Benutzern oder Aufrufern zur Verfügung?

Syntax:

Signatur von Methoden, Typen von Parametern, Daten-/Dateistrukturen

Semantik:

Welche Auswirkung hat die Nutzung/der Aufruf dieser Schnittstelle? Welche Daten oder Zustände ändern sich? Wahrnehmbare Nebenwirkungen?

Protokoll:

Detaillierter Ablauf bei Nutzung dieser Schnittstelle. Ggfs. Verweis auf Laufzeitsicht.

Restriktionen: Einschränkungen bei der Nutzung (Rechte, zeitliche Einschränkungen o.ä.)

Fehlerszenarien: Verhalten im Fehlerfall oder Ausnahmesituation

Variabilität:

Konfigurierbarkeit, Varianten oder Alternativen der Schnittstelle, hinsichtlich Syntax, Semantik, Protokoll oder anderen Eigenschaften.

Qualitätseigenschaften: (neudeutsch: Service Level Agreements)

Entwurfsentscheidungen oder Entwurfsalternativen

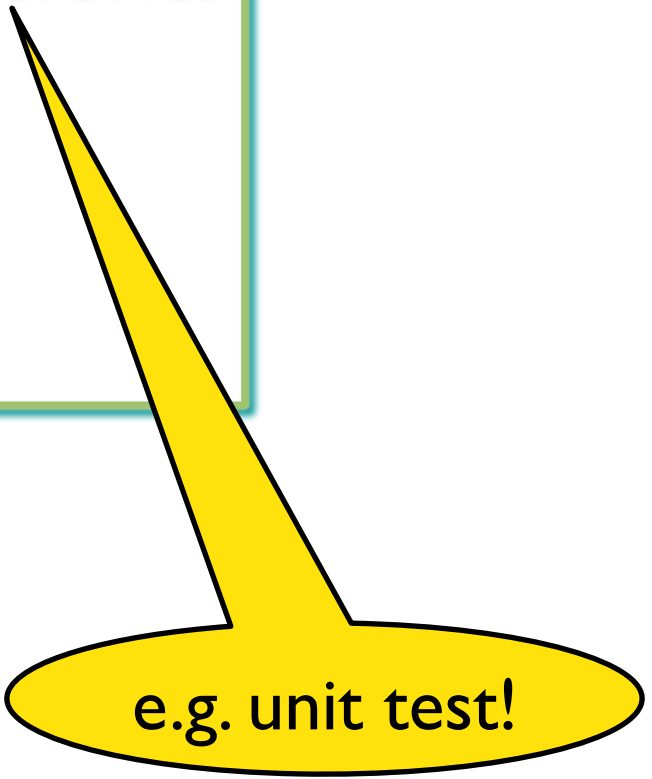
Beispieldaten

Summary: Documenting interfaces

Give small sample of a client.

Give more samples.

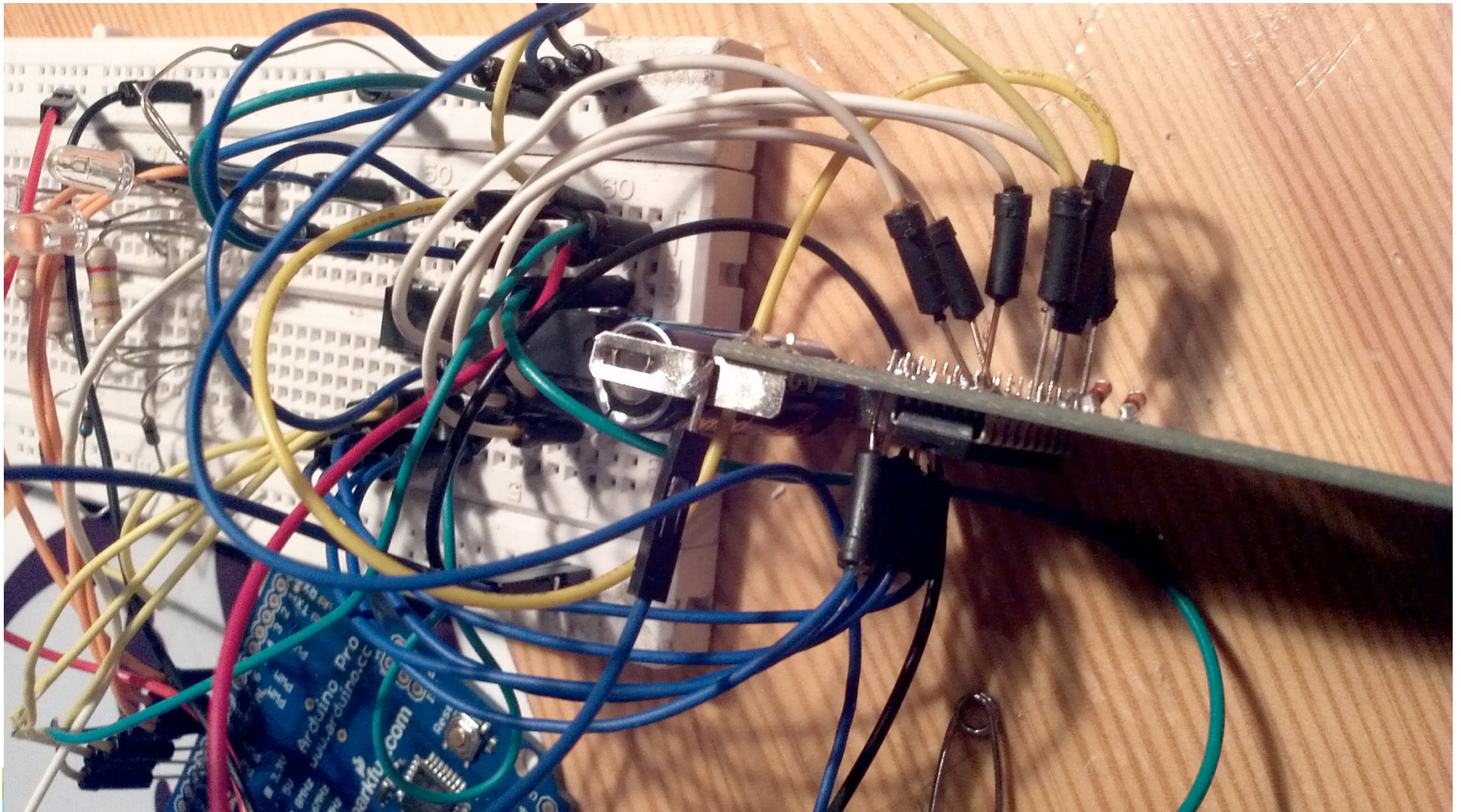
Provide reference...



e.g. unit test!

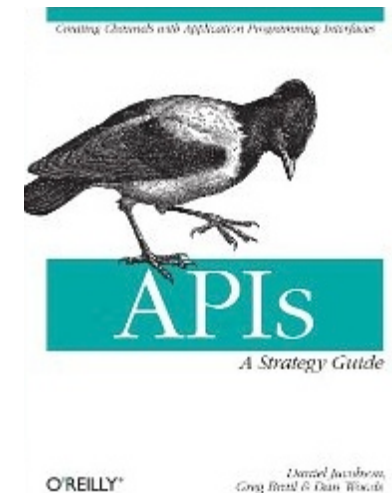
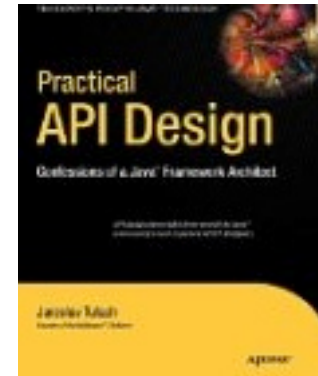
THANX

 Follow @gernotstarke



References (1)

- ▶ Joshua Bloch: API Design and Why it matters?
<http://lcsdo5.cs.tamu.edu/slides/keynote.pdf>
- ▶ J. Tulach (Netbeans-Architect)
Practical API Design
- ▶ Jasmin Blanchett: Little Manual of API Design
<http://www4.in.tum.de/~blanchet/api-design.pdf>
- ▶ D.Jacobsen et. al: APIs – A Strategy Guide.



- ▶ Jaroslav Tulach: 20 API Paradoxes (kindle e-Book), 2012

References (2)

- ▶ Bernd Schiffer (heise developer):
Flexibel mit dem Fluent Interface
<http://heise.de/-227052>
- ▶ Martin Fowler, Eric Evans: Fluent Interface
<http://martinfowler.com/bliki/FluentInterface.html>