A Test Bench Facility in Plex

Need, Design, Use

Contents

- Introduction
- An Online Test Facility for Development
- Repeatable Testing by Call Specifications
- Bundle and Execute Test Specifications
- Organizing the Test Portfolio
- Ideas, Thoughts and Experience

Soft Design

INTRODUCTION

About the Speaker

- Morten Knudsen
- Danish Post IT Department
- M.Sc. Computer Science
- Zurich Insurance
- Soft Design (Websydian Development)
- KODA – Head of IT development
- Various…
- Soft Design (Project leader, Consultant)

About the SIF Project Using the Test Bench

- Insurance application built from scratch
- 7 Plex development models
- 20 Plex developers
- SOA approach – focus on server functionality
- Online synchronization with existing system
- Soft Design is a sub-contractor (5-7 consultants)

- Funding of test bench facility…

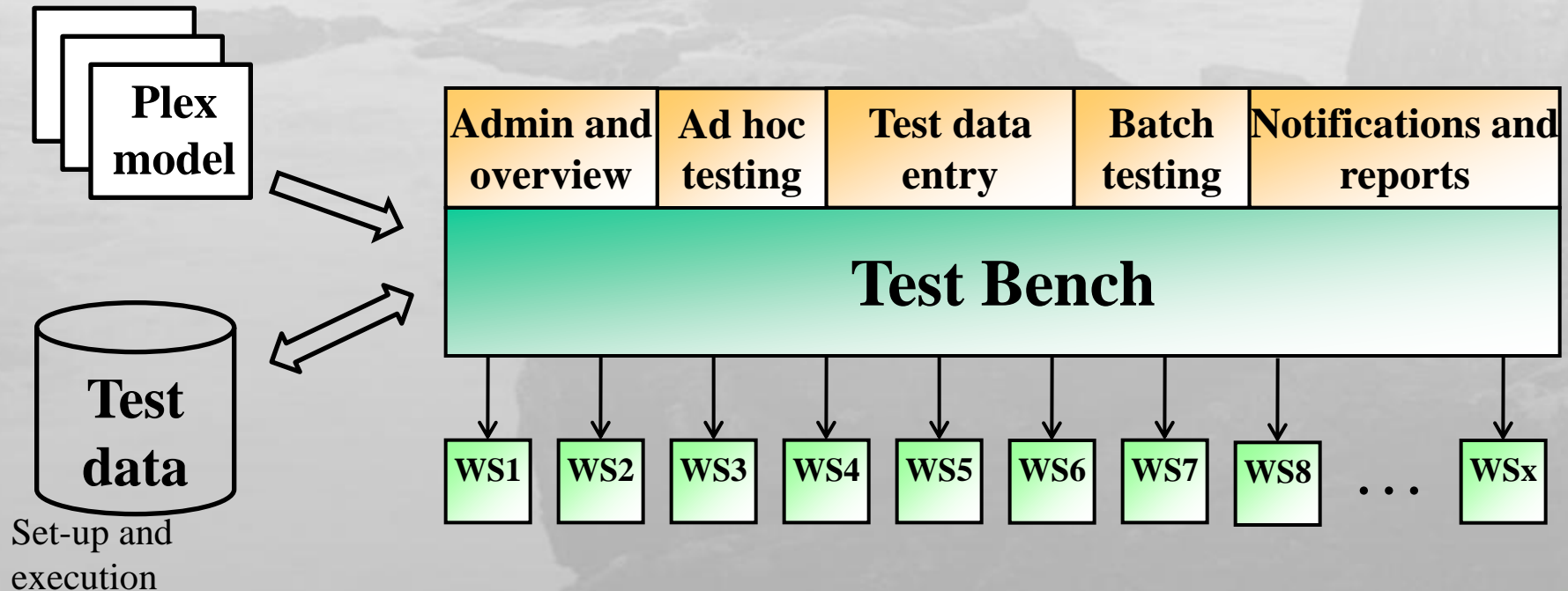Soft Design

Test Approach for SOA Architecture

- Client

- Server

- Database
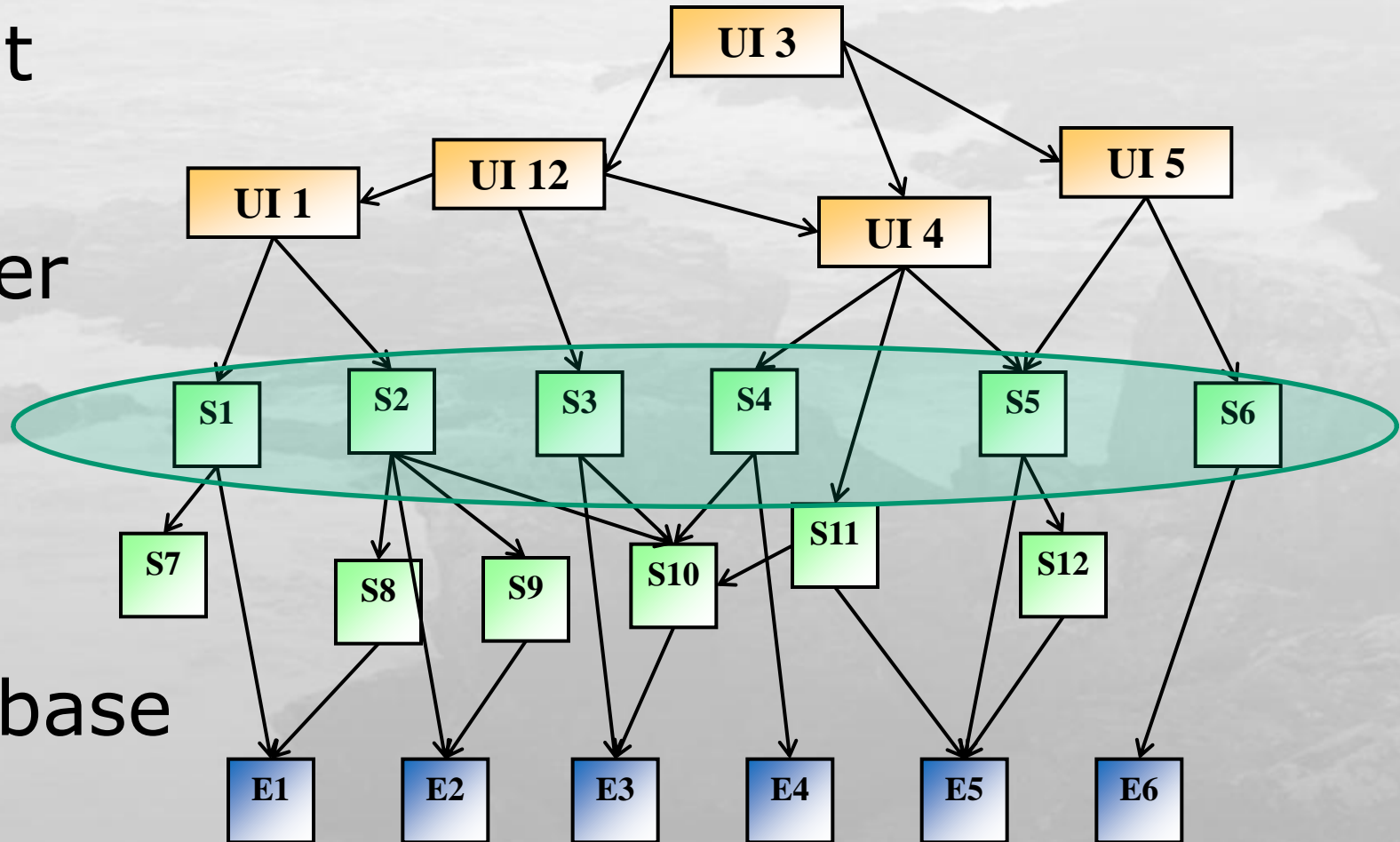
SIF application regarded as set of services

Soft Design

Test Bench Overview

- Test bench facility used by development and by test group

ONLINE TEST FACILITY FOR DEVELOPMENT

Soft Design

ca smart

Problem…

- "I want to test my function"

- I must specify a test UI
  - Capture input fields
  - Call function
  - Display output
- It takes time!
- My test data are not saved

Input

Fld A   A10

Fld B   Benny Hansen

Fld C   S1

Execute

Output

Fld X   ACC_DEP

Fld Y   Accounting

Fld Z   S1_02

Fld V   Active

Solution: Test Bench Facility

- Fast access to trial-and-error test during development
  - Register function once
- Generate test panel/page based on function specification
  - Dynamic call to selected function
  - Display returned output based on function specification
- Save results of test calls
  - For later reference and use

Call Function from Test Bench

TestBench Wrappers Functions

- Specification of *Test wrapper* for each tested function

- Same parameter interface for all wrapper

- Each wrapper 'knows' the interface of its function

- Register once!

Wrap	Wrap	Wrap	Wrap
Service S1	Service S2	Service S3	Funktion F1

ca smart
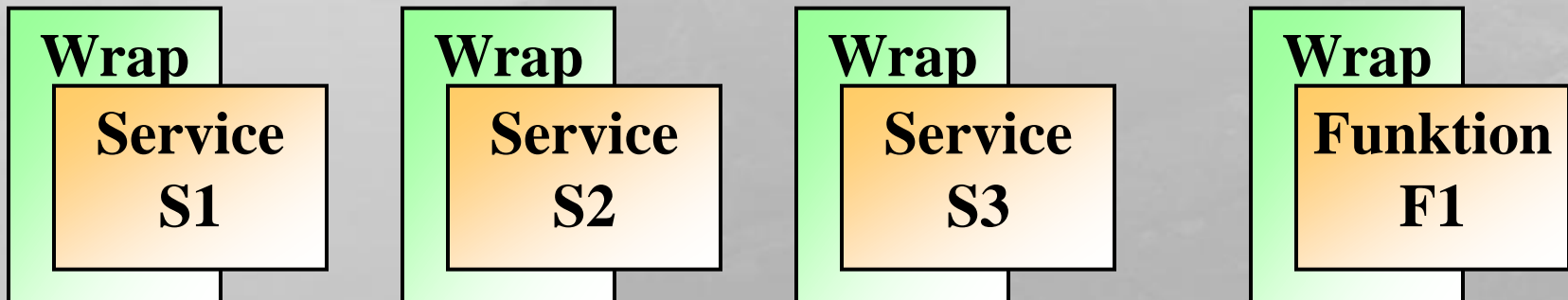
Register Function in Test Bench by Wizard

- Based on *FunctionInTestBench* abstract function
- Inherit scoped *TestBench* function
- Automatically specification of action diagram statements in inherited *TestBench* function
- Generate, build, and execute scoped *TestBench* function

Registration of Function in Test Bench

Repeat proces if updating the parameter interface of a function already registrated in the Test bench

Registration of Function in Test Bench

Test Bench Data Model 1

- ## Repository data (generated from Plex model)
  - Functions
  - Parameter fields
- ## Test bench data
  - Performed calls
  - Registered values

Test Bench   **Performed**

| Function | ← | Performed call |

| Field | ← | Parameter field | ← | Registered value |

Soft Design
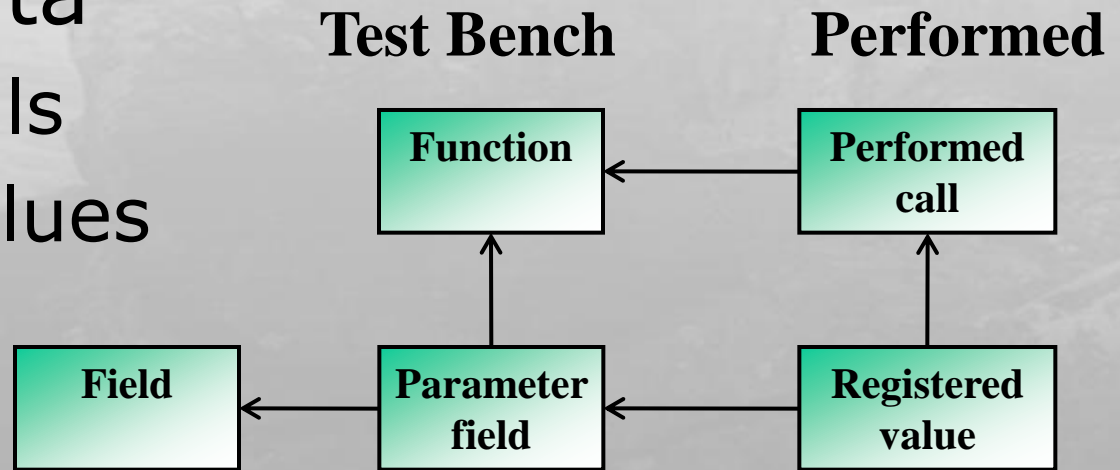
Returned status as Output Field

Display performed call

Sequence:	
Call spec text:	
Function impl. name:	POf3F
Full function name:	POL Police.Fetch.SF oplysninger til begæring
Start date:	18-05-2011
Start time:	113626
Sequence:	3591

[Refresh] [Show messages] [Edit and run]

Field use	GroupBy ▲	Impl. name	Full object name
Input	Input_FetchKey(001)	POLSekv	POL.Sekvens
Input	Input_FetchKey(001)	POLVers	POL.Version
Output	Output_Environment(001)	1610612754	*Returned status
Output	Output_FetchedData(001)	PPFOF	POL.Hovedforfalds måned
Output	Output_FetchedData(001)	POLARS	POL.Årsagskode
Output	Output_FetchedData(001)	POLOPAS	POL.Opsig i nuværende sel:
Output	Output_FetchedData(001)	NANUM	_
Output	Output_FetchedData(001)	PIBRC	BGR.Branchegruppe
Output	Output_FetchedData(001)	PIPOL	PFD.Policenummer
Output	Output_FetchedData(001)	BSBRC	BRK.Branchenr
Output	Output_FetchedData(001)	PPDNbr	PPD.Produktnummer
Output	Output_FetchedData(001)	TFSTK	PFG.Stempelkode
Output	Output_FetchedData(001)	TFSTB	PFG.StempelafgTFSTB

Show interface

[LoadGridData]

Field use...	Variable	Impl. name	Full object name	Ty.	L...	Co.	Oc...	Is dual
GroupBy: I_Input(1)								
Input	Input	BSBRC	BRK.Branchenr	Character	2	Text	1	
Input	Input	PPDNbr	PPD.Produktnum...	Numeric	5	Text	1	
Input	Input	PGTKode	PGT.Genstands...	Numeric	5	Text	1	
Input	Input	AntalGen	_Work.Antal dæ...	Numeric	9	Text	1	
GroupBy: I_UpdateData(1)								
Input	UpdateData	PFMStUDt	PFM.Studieophø...	Numeric	8	Text	1	
Input	UpdateData	PFMTagTy	PFM.Tagbelægni...	Character	1	Combo	1	
Input	UpdateData	PFMFSted	PFM.Forsikrings...	Character	37	Text	1	
Input	UpdateData	T17KOD	_	Numeric	4	*Not ch...	1	
GroupBy: I_UpdateKey(1)								
Input	UpdateKey	POLSekv	POL.Sekvens	Numeric	15	Text	1	
Input	UpdateKey	POLVers	POL.Version	Numeric	9	Text	1	
Input	UpdateKey	PGESeq	PGE.Sekvens	Numeric	6	Text	1	
GroupBy: O_Environment(1)								
Output	Environment	1610612754	*Returned status	Numeric	7	Text	1	

Soft Design

ca smart

Dual Field Parameters

Dual Field Parameters

- Input value and output value saved for dual fields

Arra
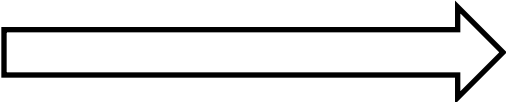
Show interface

| Function impl. name: | SY2hF |
| Full function name: | FLD Field.Fetch.BF wrappe |

[LoadGridData]

Field use type	Variable	Impl. name	Full
Dual	Control	352321596	Row

GroupBy: I_Control(1)

| Input | Control | 352321583 | Posi |
| Input | Control | WSYDRRN | Rela |

GroupBy: I_Input(1)

| Input | Input | FLDNmCon | Field |

GroupBy: I_Position(1)

| Input | Position | TFIImplN | FLD |

GroupBy: O_Environment(1)

| Output | Environment | 1610612754 | *Ret |

GroupBy: O_FetchedData(64)

Output	FetchedData	WSYDRRN	Rela
Output	FetchedData	TFIImplN	FLD.
Output	FetchedData	TMOCode	Mode
Output	FetchedData	FLDObjNm	Full
Output	FetchedData	FLDType	Field
Output	FetchedData	FLDLen	Field
Output	FetchedData	FLDLenDp	Field
Output	FetchedData	FLDContr	Field
Output	FetchedData	FLDLabel	Field
Output	FetchedData	FLDTopLb	Top label used

Display performed call

Sequence:	10
Call spec text:	
Function impl. name:	SY2hF
Full function name:	FLD Field.Fetch.BF wrapper
Start date:	21-03-2011
Start time:	151630
Sequence:	695

[Refresh] [Show messages] [Edit and run]

Field use t...	Variable	Oc...	Impl. name	Full object name	L...	Ty.	Value
Output	FetchedData	24	FLDObjNm	Full object name	0	Character	Køretøjets aksel antal
Output	FetchedData	24	FLDTopLb	Top label used	0	Character	Køretøjets aksel antal
Output	FetchedData	24	FLDType	Field type	0	Character	N
Output	FetchedData	24	TFIImplN	FLD.Field impl name	12	Character	GEBAKSTK
Output	FetchedData	24	TMOCode	Model code	0	Character	
Output	FetchedData	24	WSYDRRN	RelativeRecordNumber	15	Numeric	25
Output	FetchedData	25	FLDContr	Field control	0	Character	
Output	FetchedData	25	FLDLabel	Field label used	0	Character	CRM anvendelses kode
Output	FetchedData	25	FLDLen	Field length	0	Numeric	0
Output	FetchedData	25	FLDLenDp	Field display length	0	Numeric	0
Output	FetchedData	25	FLDObjNm	Full object name	0	Character	CRM anvendelses kode
Output	FetchedData	25	FLDTopLb	Top label used	0	Character	CRM anvendelses kode
Output	FetchedData	25	FLDType	Field type	0	Character	N
Output	FetchedData	25	TFIImplN	FLD.Field impl name	12	Character	GEBANVKD
Output	FetchedData	25	TMOCode	Model code	0	Character	
Output	FetchedData	25	WSYDRRN	RelativeRecordNumber	15	Numeric	26

| | | | Field label used | Character | 0 | | 64 |
| | | | Top label used | Character | 0 | | 64 |

TestBench Functionality

- ## Registration of function
  - Create *Function*, *Field*, and *Parameter field* records in Repository

- ## Execute call of wrapped function to be tested
  - Create *Performed call* record
  - Create *Registrated value* records for input values
    - Based on Parameter field repository data

  -

  TestBench

  My function

  - Retrieve *Registrated parameter value* records
  - Get start timestamp
  - Call wrapped function
    - Map with retrieved input parameters
  - Get end timestamp
  - Save returned output/dual values as *Registered value* records
  - Update *Performed call* record with start and end timestamp

Soft Design

ca smart

REPEATABLE TESTING BY CALL SPECIFICATIONS

Soft Design

ca smart
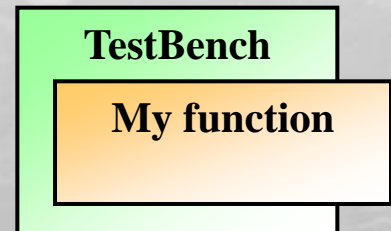
Call Specifications

- ## What is a Call specification?
  - Declaration of parsed input and expected output
  - Associated to function and described by a number of attribute

- ## Why have Call specifications?
  - Explicit definition of expected output for selected output fields
  - Establish library of relevant calls for each function
  - Specification of various parameters and information controlling/describing call
  - Basis for bundled test runs…

Call Specifications & Performed Calls

Call Specifications

- Description of call
- Responsible user
- Call control information
- Used input
- Selected output
- Expected output and max duration
- Pointer to representing Performed call…
- Various attributes associate to bundling

Performed Calls

- Information about performed call
  - Who, when, time taken
- Received input
- Generated output
- Derivation from expected output

Test Bench Data Model 2

Before:

Repository **Performed**

```
Function  ← Performed call
   ↑            ↑
Field ← Parameter field ← Registered value
```

After:

Repository **Specification** **Performed**

```
Function ← Call specification ← Performed call
   ↑            ↑                    ↑
Field ← Parameter field ← Specified value ← Registered value ← Recorded error
```

Soft Design
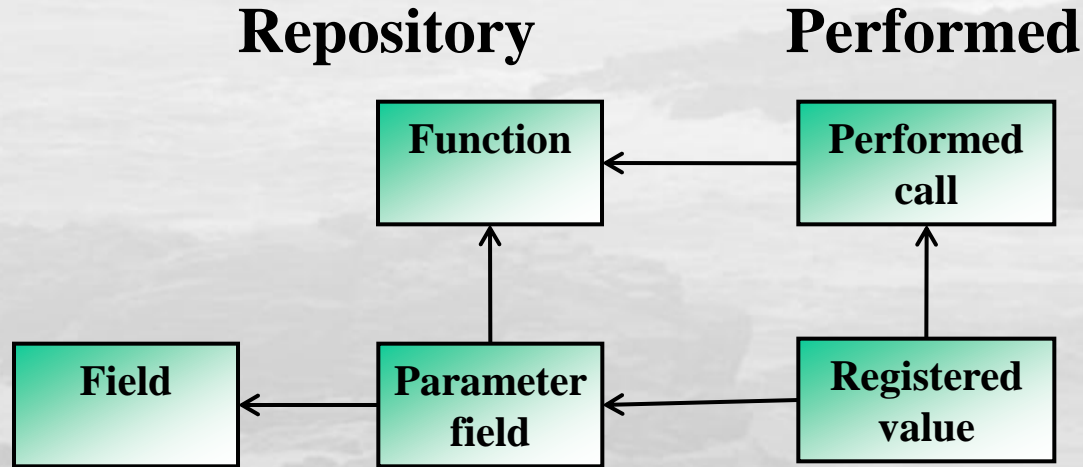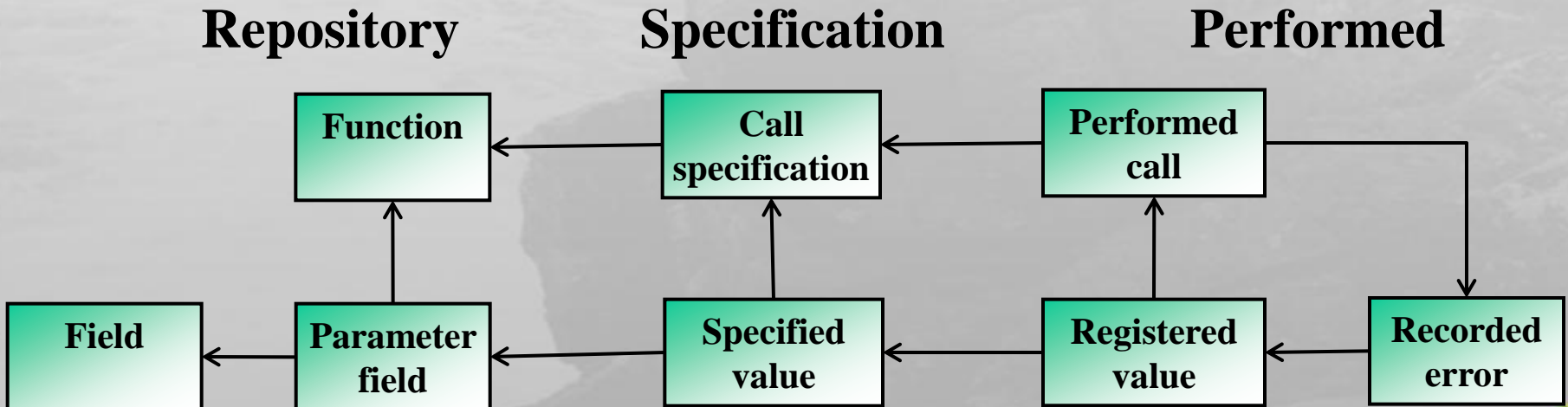
Call Specifications & Performed Calls

- Decision: Separate entities
- Copy, Copy parameters, Execute/run, Define as

Copy

Edit and run

Execute

Call specification

Performed call

Update spec

Create as spec

Copy parm from

Soft Design

ca smart
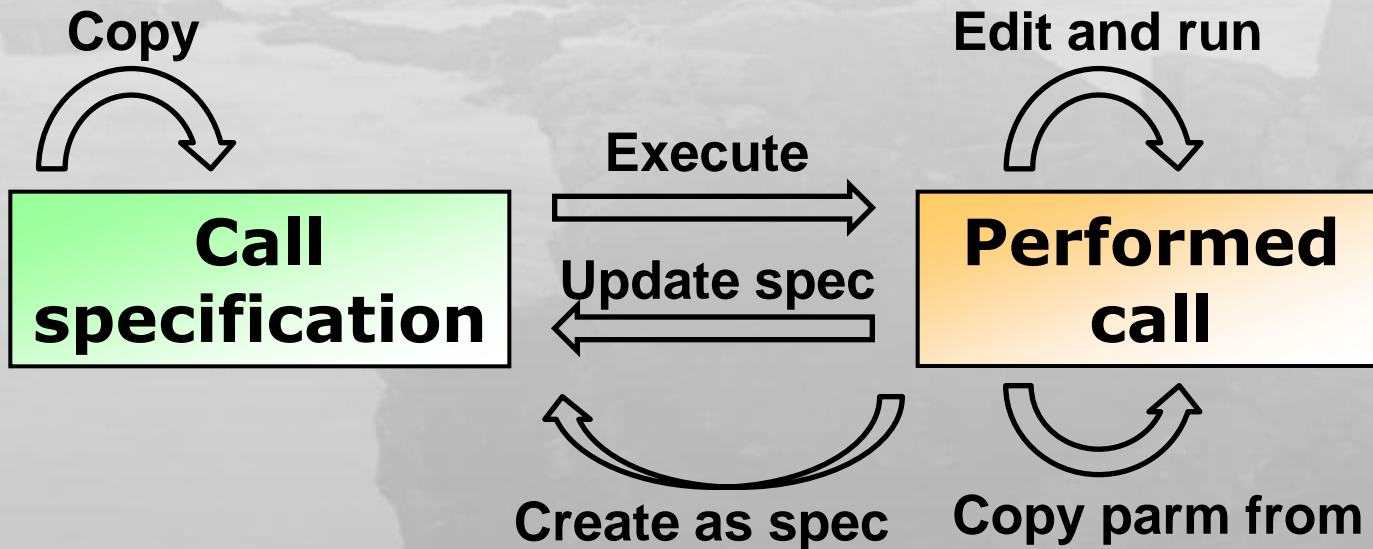
Specify Values by Reference
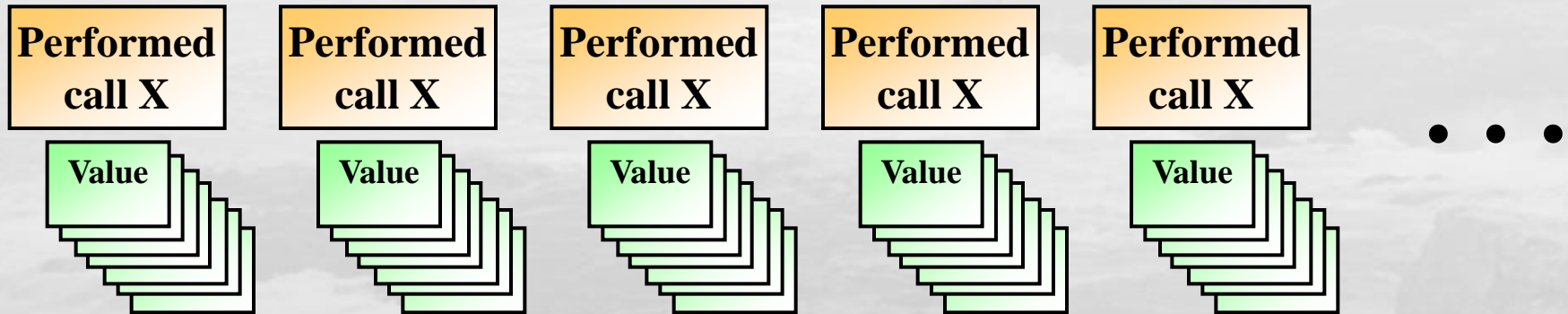
Reference to identical performed call:
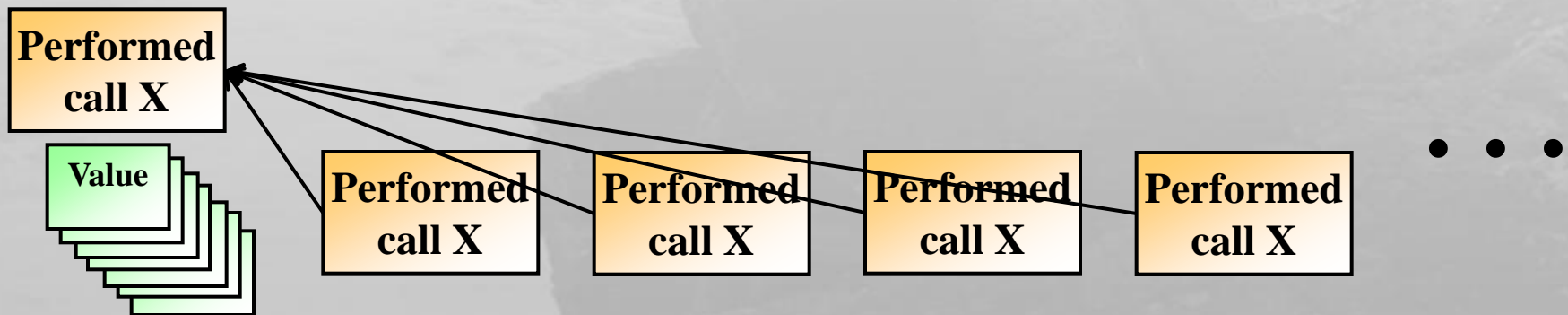
Specify Values by Reference

- *Call specification* may point to *Performed call*
  - Representing input and expected output
  - New *Performed call* points to referred *Performed call* to represent values
  - Clear reference when *Call specification* changed
  - Save disk space and I/O at later bundling
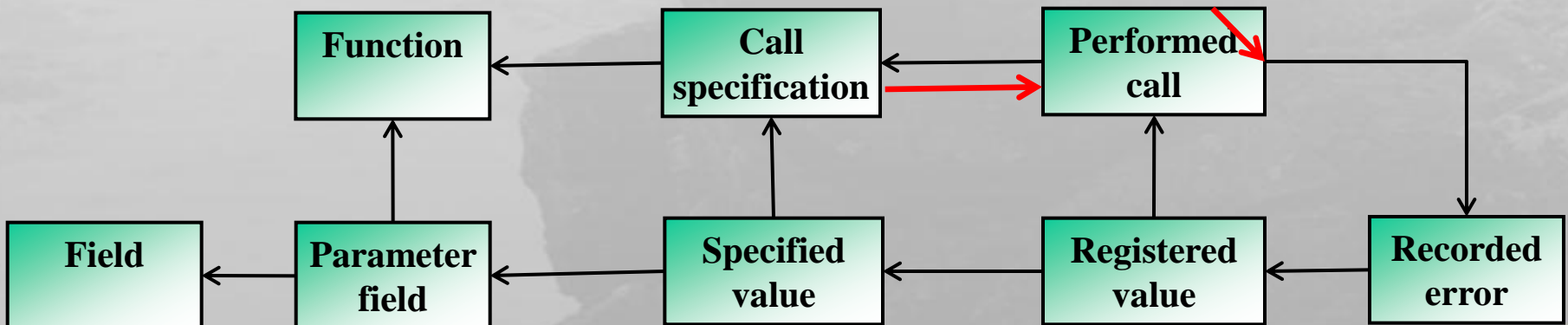
BUNDLE AND EXECUTE TEST SPECIFICATIONS

Bundling Facility to be Used by Test Group

- Automatic support of repeatable execution of specified test
  - Batch and online
- Documentation of result of test runs
  - Timestamps and duration
  - Actual output differs from specified/expected output
- Administration and overview of bundle specifications

Test Bench Overview

- Test bench facility used by development and by test group

- Register functions by test wrappers

  - Hereafter all test and administration performed through Test bench

Admin and overview	Ad hoc testing	Test data entry	Batch testing	Notifications and reports
Test Bench				

Plex model

Test data

Set-up and execution

WS1 WS2 WS3 WS4 WS5 WS6 WS7 WS8 ... WSx
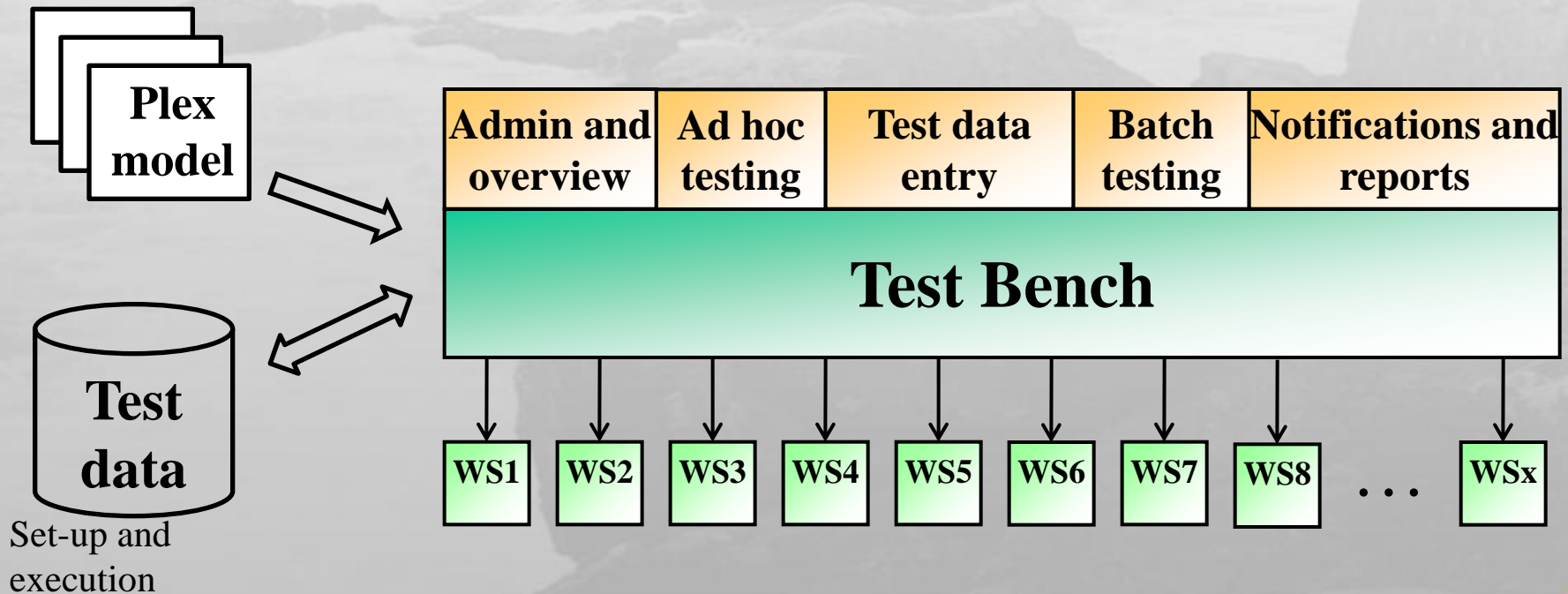
Organisation in a 4-Levels Hierarchy

- Model
- Suite
- Run
- Pack
- Call specification

- Lower levels can be run individually

Soft Design

Execution of Pack/Run/Suite

Execution of Suite

Execution of Run

Execution of Pack

Function A

Function B

Function C

Function D
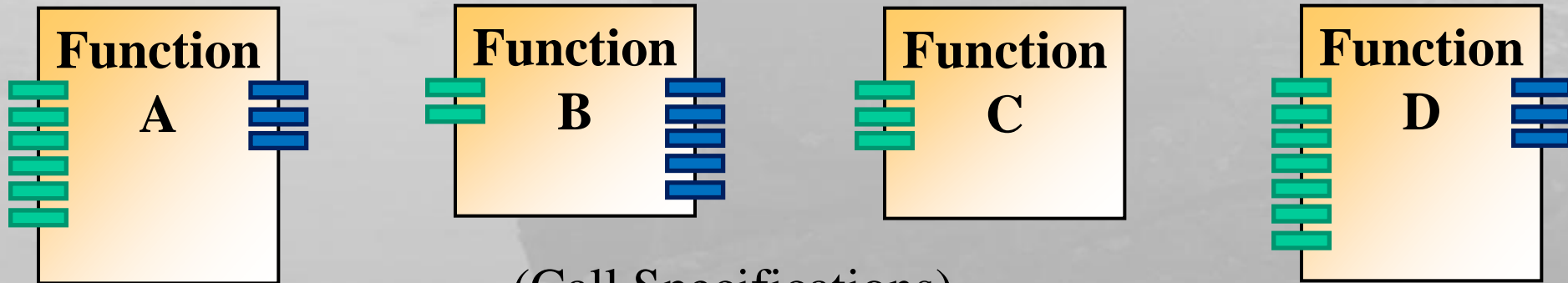
(Call Specifications)

Test Bench Data Model 3

Specification Data & Performed Data

- Specification data
  - Structure (in hierarchy)
  - Descriptions
  - Input values and expected output values
  - Call settings…
- Performed data
  - Structure (in hierarchy)
  - Actual values
  - Statistics/counts, timestamps/duration

Additional Requirements…

- Use Previous Output as Input
  - More dynamic less rigid test specifications
- Check Functions
  - Validation of database updates
- Test Auxillary Functions
  - Matter of definition

Use Previous Output as Input

- Output from function call to be used as input for later function call(s)
  - E.g. surrogate, current date
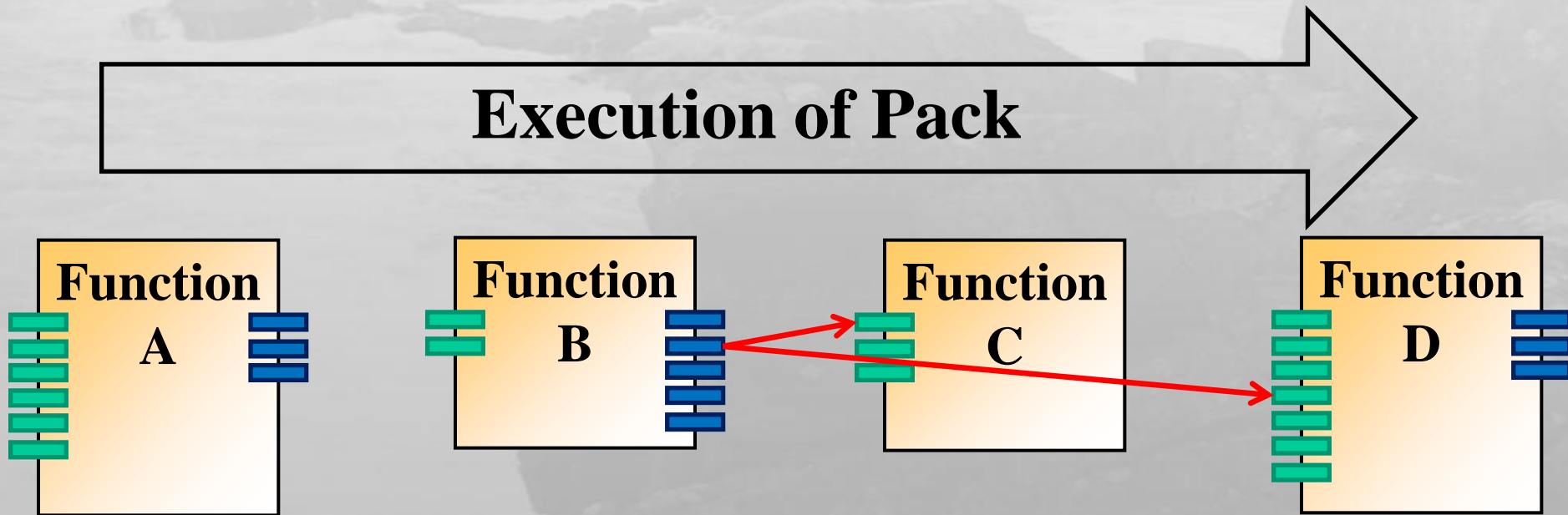
Use Previous Output as Input

- Scope: With pack/run
- Map *Input reference* to *Referred output*
- 4 types of Specified values
  - Input value
  - Input reference
  - Expected output
  - Referred output

More dynamic less rigid – test specifications do not need to rely on specific data values

Soft Design

ca
smart

Check Functions

- Test update of database
  - Look up data values
- Report errors on 'main' function
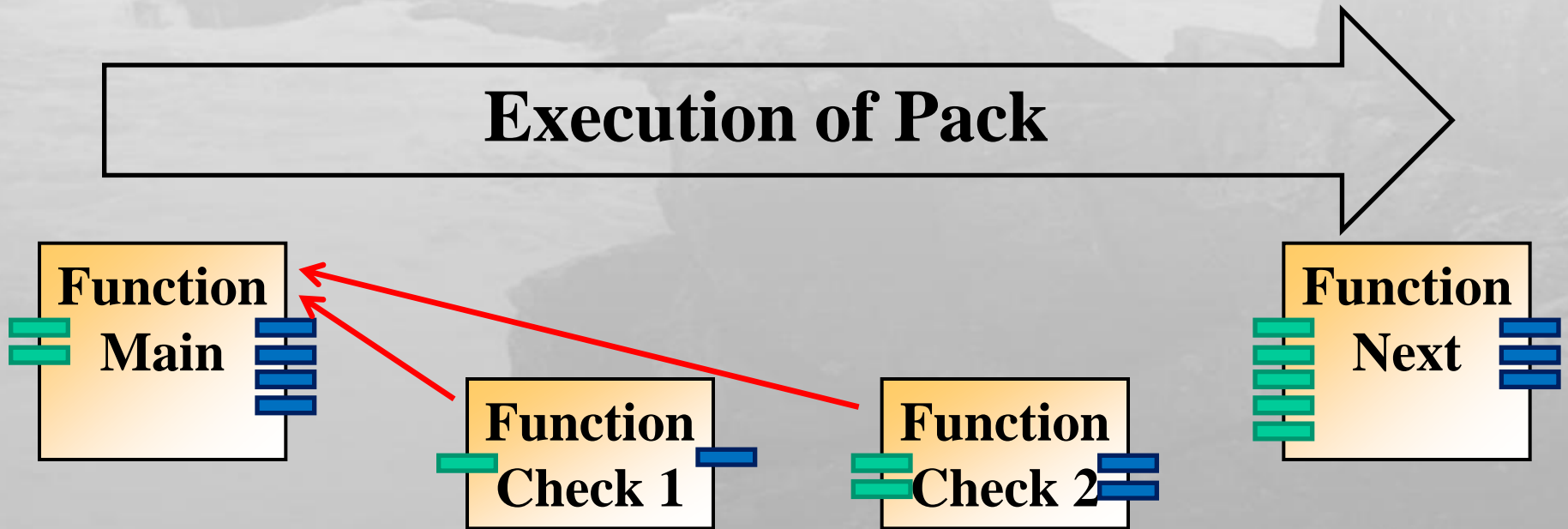
Test Bench Data Model 4
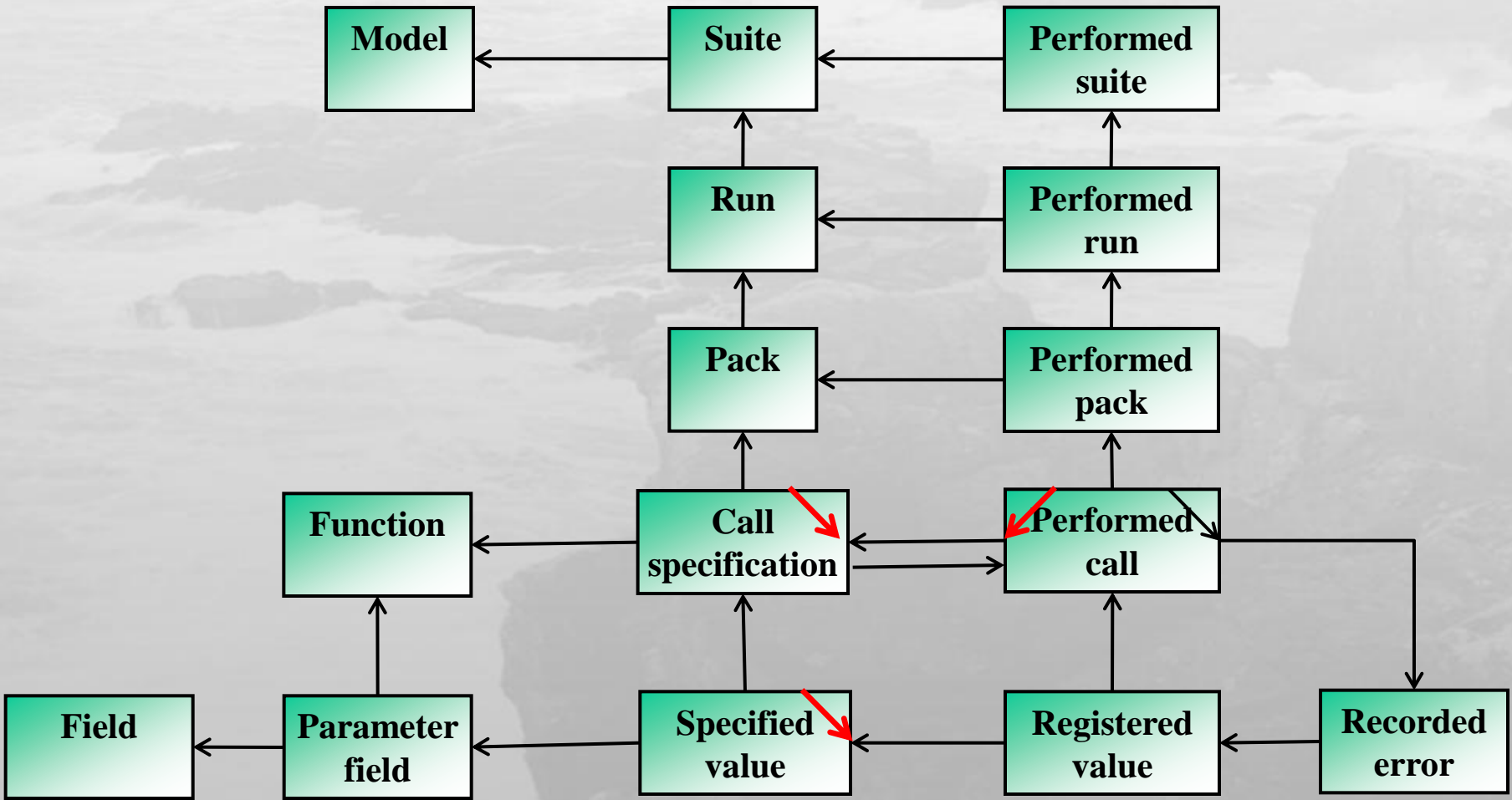
Test Auxillary Functions

- Not part of application
  - Purpose only to support test
- Different types
  - Creation of reference data
    - E.g. create header record for sub record testing
  - Fast retrieval of relevant data
    - Get current date, Check functions
  - Clean-up
    - Before/after run

Other Features/Settings

- Call Spec Status (Active, Construction)
- Sorting
- Exit on error
- Save all/specified
- Various statistics and counts
- Short-cuts for copying and specification of parameters
- Resolve error message list…

ORGANIZING THE TEST PORTFOLIO

How to Specify and Organize the Test Portfolio

- Focus
  - All functions, Update functions, Rules, Calculations, **Services**
- Structure
  - Use cases, **Data**, Functionality

Plex model

Test data

Set-up and execution

Admin and overview	Ad hoc testing	Test data entry	Batch testing	Notifications and reports

Test Bench

WS1 WS2 WS3 WS4 WS5 WS6 WS7 WS8 … WSx
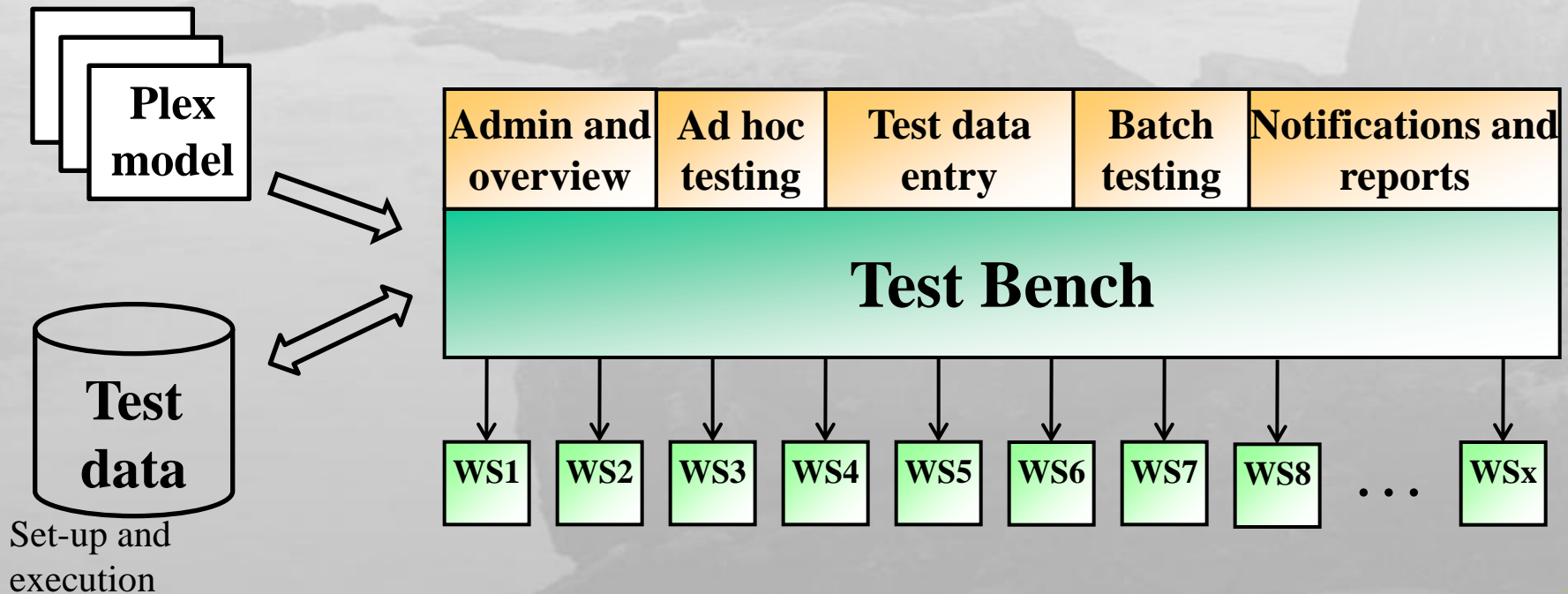
SOA Approach to SIF Application

- Client

- Server

- Database
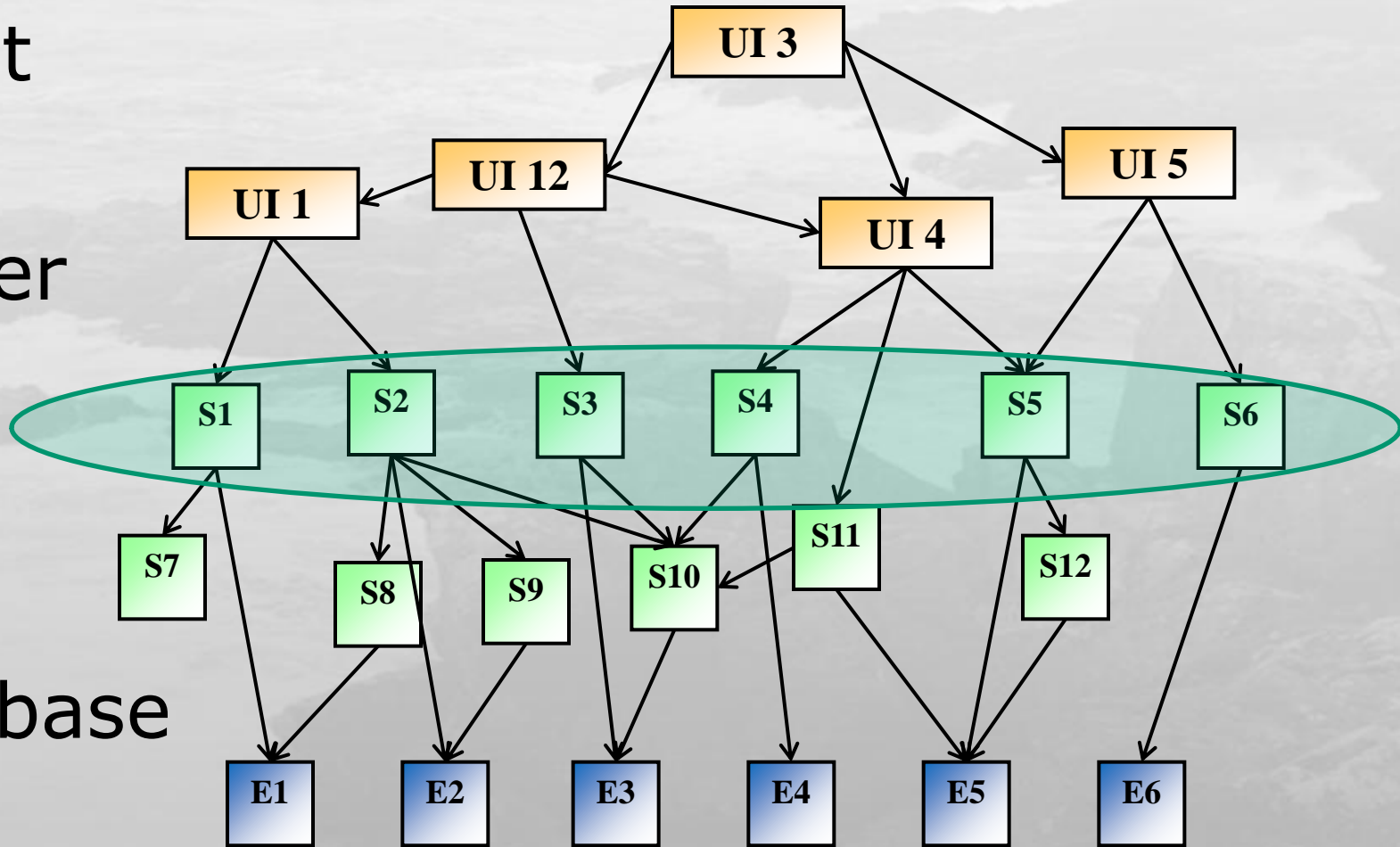
SIF application regarded as set of services

Focus of Test Specifications

- Functions (services) accessed externally or by user interface
  - Transactions
  - Data retrieval
- Important functions/components
  - Calculations
  - Derivation of data
  - Abstractions…

Data-Oriented Organization of Test Portfolio

- Functionality structured by data in Plex
  - Functions scoped by entities

- Scope test portfolio by data model
  - Model: Plex model
  - Suite: Subject area
  - Run: Entity
  - Pack: Function category…

Validation Rules Associated to Data and Transactions

Client

Server

Event handler → Transaction

Info at error (handle)

1. Data validation
2. Function validation
3. Validation in transaction
4. Perform transaction
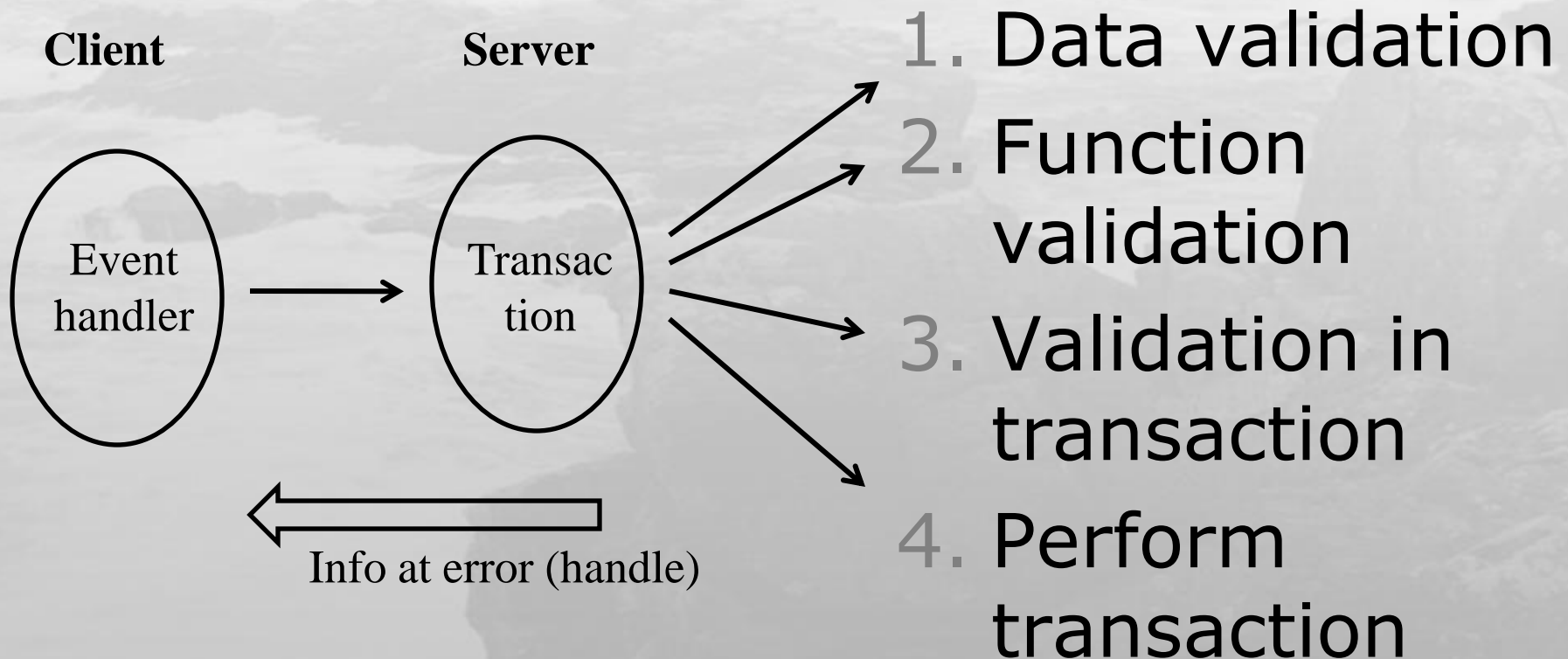
Server-Side Validation and Message Generation

1) Call transaction for Event handler
2) Call associated validations from transaction
3) Perform validation functions
4) Call message function (on error)
5) Create record for error message (on error)
6) Perform transaction
7) Call next page or call error on page
8) Retrieve error message(s) associated to *MsgID*
9) Display and mark errors in page
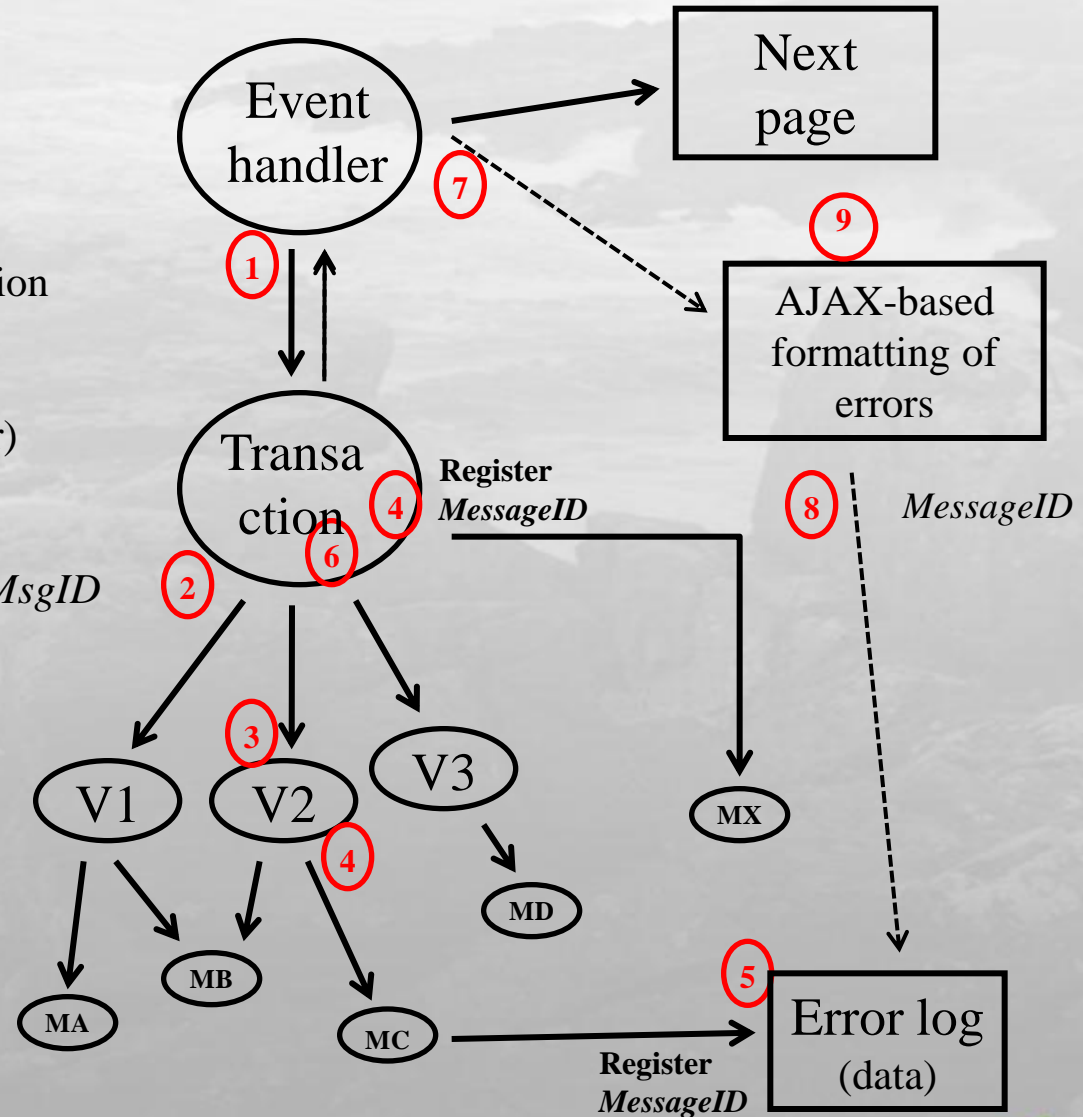
Function Categories (Packs) to Capture Important Aspects of Tests

- CRUD
  - Important retrieval functions, components for create/update/delete
- VAL
  - Validation associate to datamodel
- VALFNC
  - Triple *FNC comprises FNC* used for validation
- VALTRN
  - Validation by transaction, attachment of validations
- TRAN
  - Update of database performed by main transactions

Testing of Validation Rules

- Server-side validation
  - Error state passed back in *Returned status* as a pointer to list of messages
- Expected output as contents of list
  - Compare with actual list returned by transaction
- Error message list facilitated by Websydian Express...

Returned status

| M1 | M2 | M3 | M4 |

Function X

4 messages generated
Resulting output: M1,M2,M3,M4
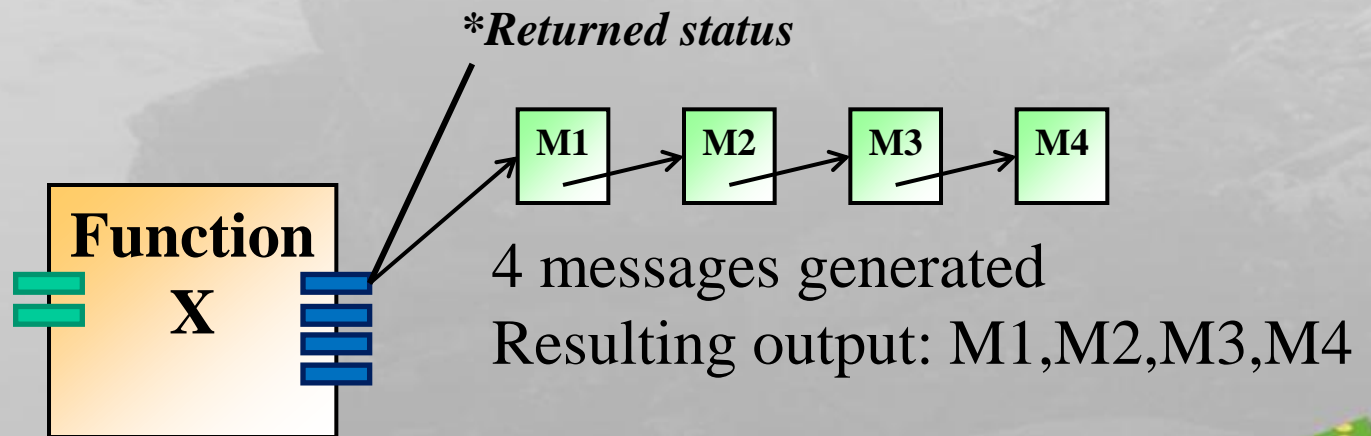
Testing of Validation Rules

Display performed call

Sequence:	438
Call spec text:	Tjek validering af børne CPR 6-10
Function impl. name:	PO14fF
Full function name:	PUL Ulykke.Update UPDCR Ulykke - børn
Start date:	19-05-2011
Start time:	130407
Sequence:	3661

[Refresh] [Show messages] [Edit and run]

Field use t...	Variable	Oc...	Impl. name	Full object name	L...	Ty.	Value	Expected value
Input	UpdateData	1	PULCpr04	PUL.CPR nummer 4	10	Numeric		
Input	UpdateData	1	PULCpr05	PUL.CPR nummer 5	10	Numeric		
Input	UpdateData	1	PULCpr06	PUL.CPR nummer 6	10	Numeric	0511670000	0511670000
Input	UpdateData	1	PULCpr07	PUL.CPR nummer 7	10	Numeric	0511670000	0511670000
Input	UpdateData	1	PULCpr08	PUL.CPR nummer 8	10	Numeric	0511670000	0511670000
Input	UpdateData	1	PULCpr09	PUL.CPR nummer 9	10	Numeric	0511670000	0511670000
Input	UpdateData	1	PULCpr10	PUL.CPR nummer 10	10	Numeric	0511670000	0511670000
Input	UpdateData	1	PULCpr11	PUL.CPR nummer 11	10	Numeric		
Input	UpdateData	1	PULCpr12	PUL.CPR nummer 12	10	Numeric		
Input	UpdateData	1	PULCpr13	PUL.CPR nummer 13	10	Numeric		
Input	UpdateData	1	PULCpr14	PUL.CPR nummer 14	10	Numeric		
Input	UpdateData	1	PULCpr15	PUL.CPR nummer 15	10	Numeric		
Input	UpdateData	1	POLAnFor	PUL.Antal forsikrede	5	Numeric	5	5
Input	UpdateKey	1	POLSekv	POL.Sekvens	15	Numeric	2005112461	2005112461
Input	UpdateKey	1	POLVers	POL.Version	9	Numeric	1	1
Input	UpdateKey	1	PGESeq	PGE.Sekvens	6	Numeric	1	1
Output	Environment	1	1610612754	*Returned status	7	Numeric	O1:PULCpr06,O1:PULCpr10,O1:PULCpr09,O1:PULCp...	O1:PULCpr06,O1:PULCpr10,O1:PULCpr09,O1:PUL...

Soft Design

ca smart

Display Application Messages

IDEAS, EXPERIENCE, AND THOUGHTS (SO FAR)

Organizational Issues

- Early focus on testing
  - Enforments of standards vs. provision of strong development facilities
- Early focus on server transactions and rules
- Integrated part of development
  - Developers responsible for test of own modules
  - Pass relevant test specifications to test group
- Run test portfolio regularly to check for unexpected side effects
  - Relevant after major tasks/changes
- Pre-condition for agile development
  - Automated testing supports frequent delievery
- Test specification portfolio Part of Delivery

Features Planned Not Yet Included

- Minimum time for execution
- Batch running of test suites
- Java implementation
- *Error call* status

Soft Design

Future Issues and Development

- Isolate Test bench in separate model
- Part of Websydian Express?
- Versions of functions and their parameter interfaces
- Handling of different environments and configurations

Test Bench Influence on Development

- Shorter test cycle
- Common point of reference
- Dead code detection and removal
- Superfluous parameters

I wish!

Soft Design

QUESTIONS ???