“There’s an app for that.”

Leaders in software testing

Testing the API behind a mobile app

System testing in a cloud enabling project

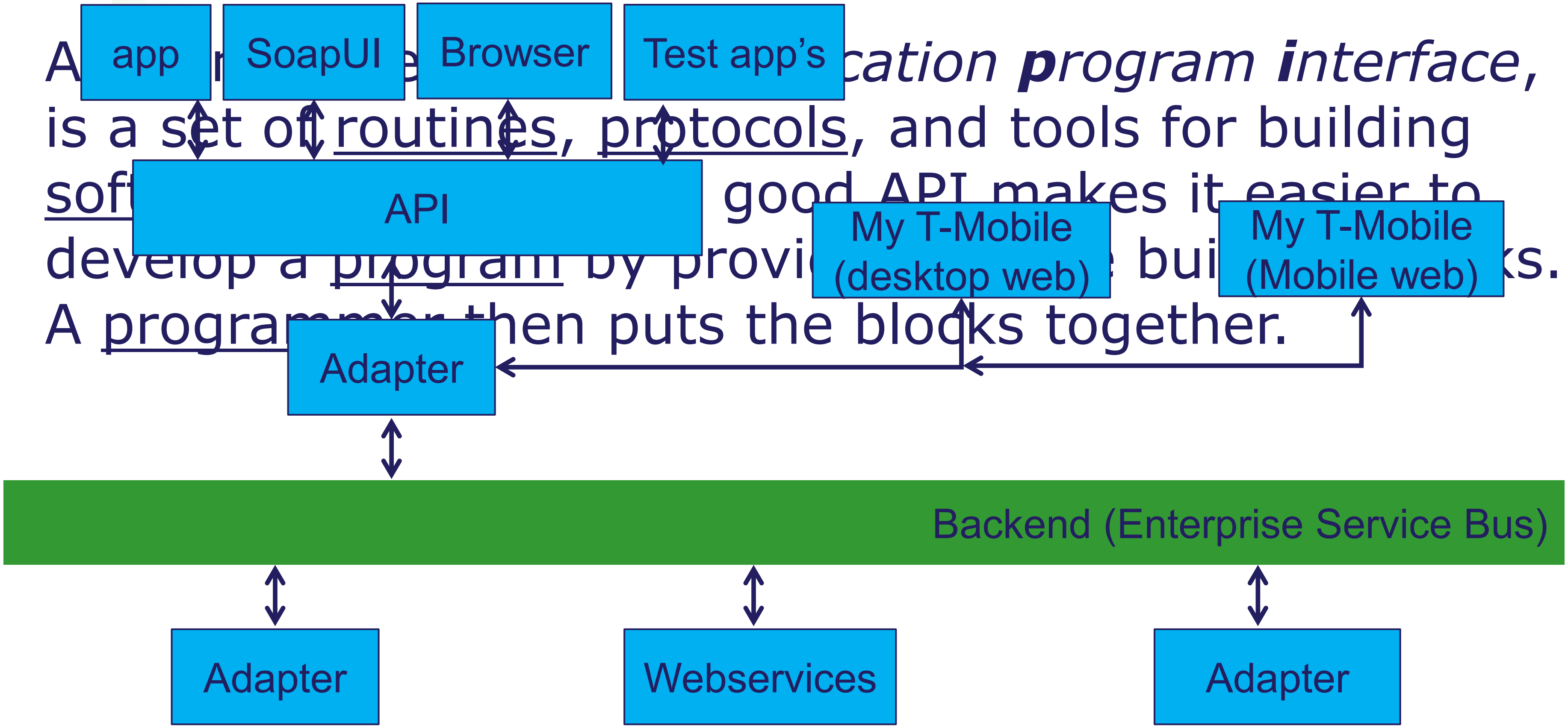
Marc van ‘t Veer

Content

- Introduction to an API
  - What is an API
  - Why an API?
  - Using the API
- How-to test an API
- Example defects
- Lessons learned

What is an API?

- A [illegible] r [illegible] e [illegible] *cation **p**rogram **i**nterface*, is a set of routines, protocols, and tools for building soft [illegible] good API makes it easier to develop a program by provi [illegible] bui [illegible] ks. A programmer then puts the blocks together.

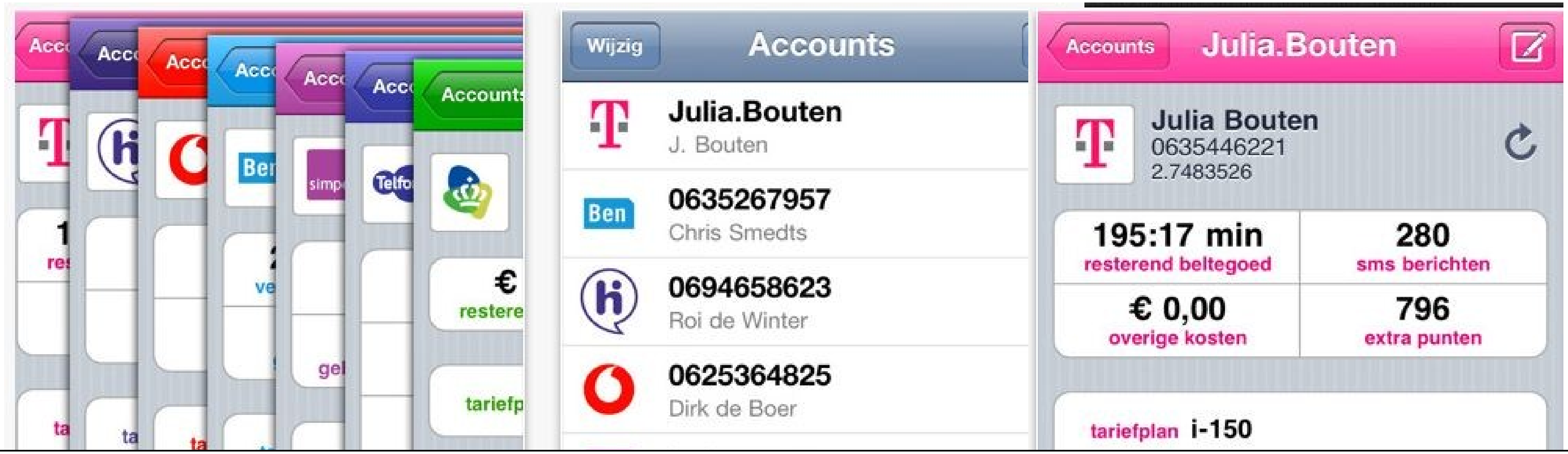


- My definition:
  – It's a framework for communication between systems
  – Own interface build upon existing internal SOA architecture

Why an API? - examples

A) Beltegoed app from Skinkies

B) Beltegoed Tufo/Zakelijk apps from D-centralize



P-Edge quotes of customers

Pointing with the blaming finger to T-Mobile *

by Vinsert

Before and after the update the app is not working. P-Edge said: "Please call customer service of T-Mobile".

But they can't help and pointing back because this app is not theirs and it's a problem of P-Edge.

Why an API? - details

- Problem
  - Too Diverse market with many platforms
  - T-Mobile is not in control of its customer contacts
  - Screen scrapers are not good and secure enough

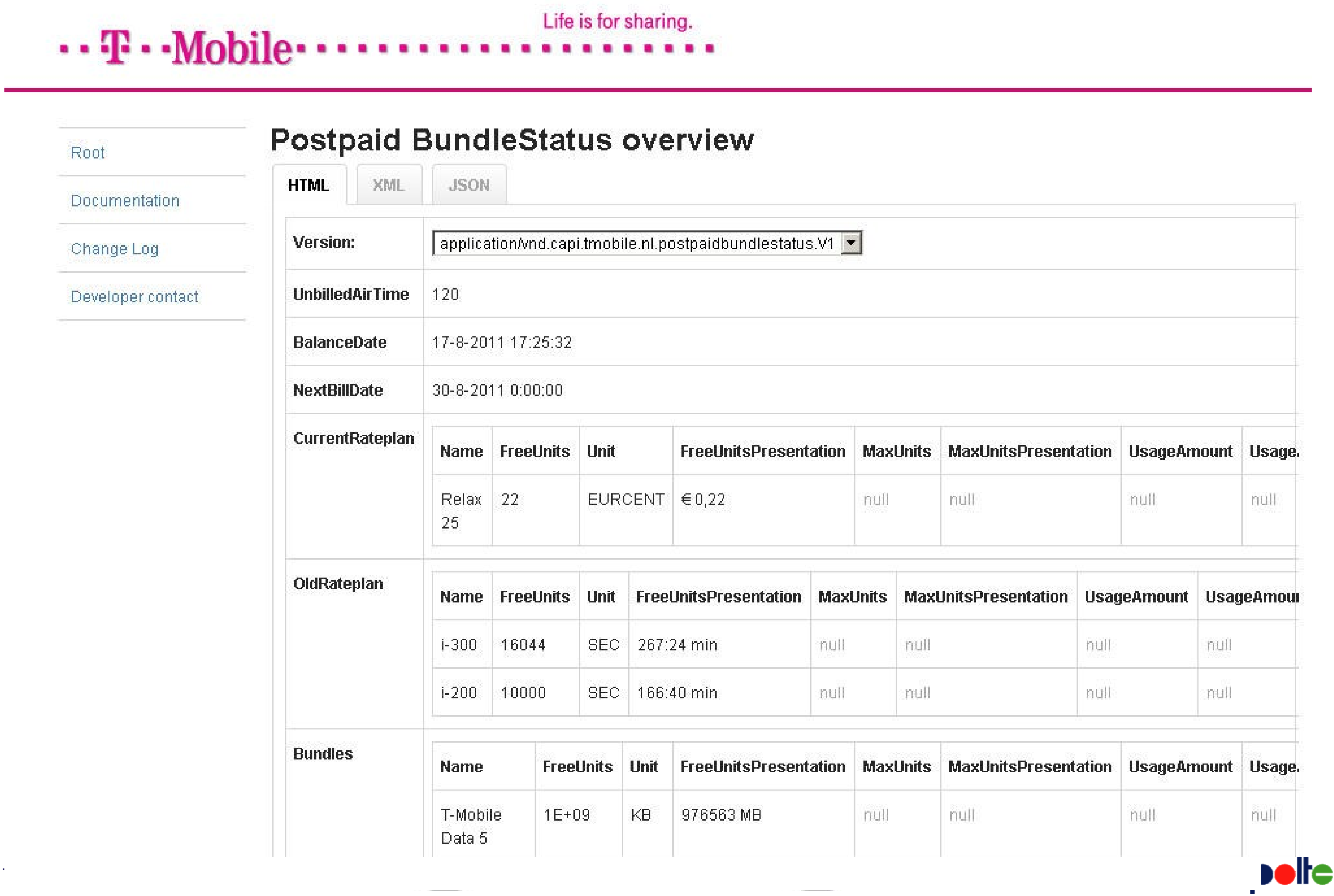
- Solution
  - Introduction of an API

- Requirements
  - Introduction must be fast, much faster than direct competitors
  - Ready for future projects

Using API - structure

NS

T-Mobile



Life is for sharing.

T-Mobile

- Root
- Documentation
- Change Log
- Developer contact

Postpaid BundleStatus overview

HTML | XML | JSON

Version:	application/vnd.capi.tmobile.nl.postpaidbundlestatus.V1
UnbilledAirTime	120
BalanceDate	17-8-2011 17:25:32
NextBillDate	30-8-2011 0:00:00

CurrentRateplan

Name	FreeUnits	Unit	FreeUnitsPresentation	MaxUnits	MaxUnitsPresentation	UsageAmount	Usage.
Relax 25	22	EURCENT	€ 0,22	null	null	null	null

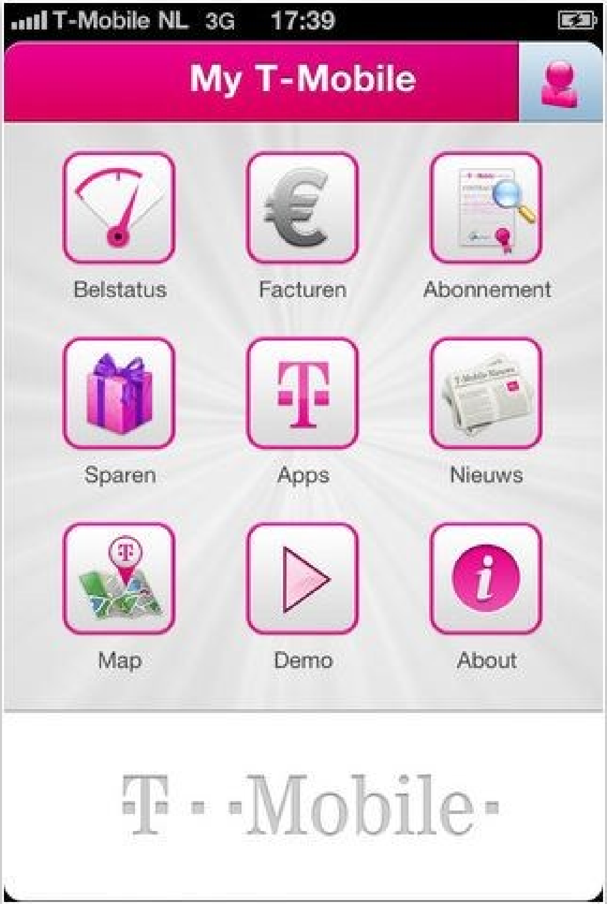
OldRateplan

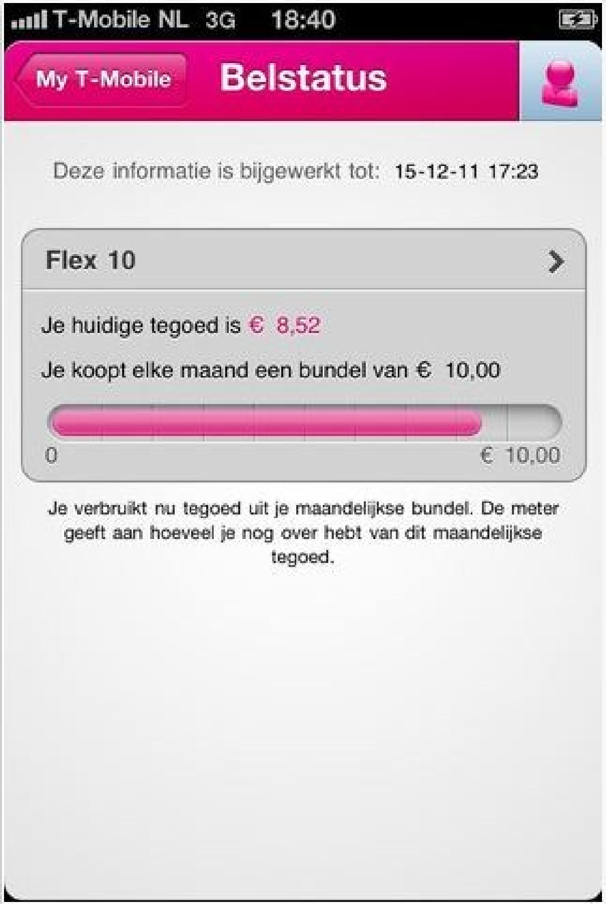
Name	FreeUnits	Unit	FreeUnitsPresentation	MaxUnits	MaxUnitsPresentation	UsageAmount	UsageAmou
i-300	16044	SEC	267:24 min	null	null	null	null
i-200	10000	SEC	166:40 min	null	null	null	null

Bundles

Name	FreeUnits	Unit	FreeUnitsPresentation	MaxUnits	MaxUnitsPresentation	UsageAmount	Usage.
T-Mobile Data 5	1E+09	KB	976563 MB	null	null	null	null

Using API – My T-Mobile app





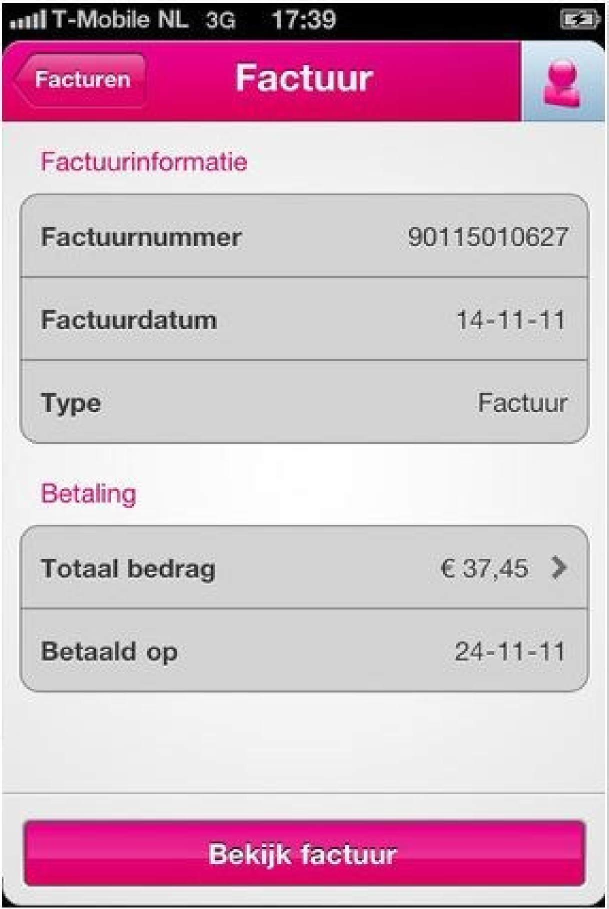
Deze informatie is bijgewerkt tot: 15-12-11 17:23

Flex 10

Je huidige tegoed is € 8,52

Je koopt elke maand een bundel van € 10,00

Je verbruikt nu tegoed uit je maandelijkse bundel. De meter geeft aan hoeveel je nog over hebt van dit maandelijkse tegoed.

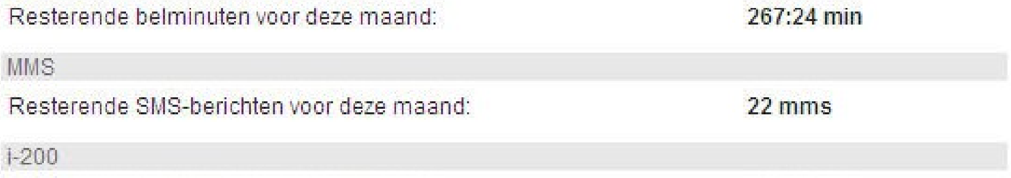


Factuurinformatie

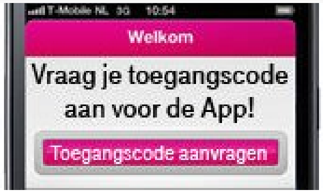
Factuurnummer	90115010627
Factuurdatum	14-11-11
Type	Factuur

Betaling

Totaal bedrag	€ 37,45
Betaald op	24-11-11

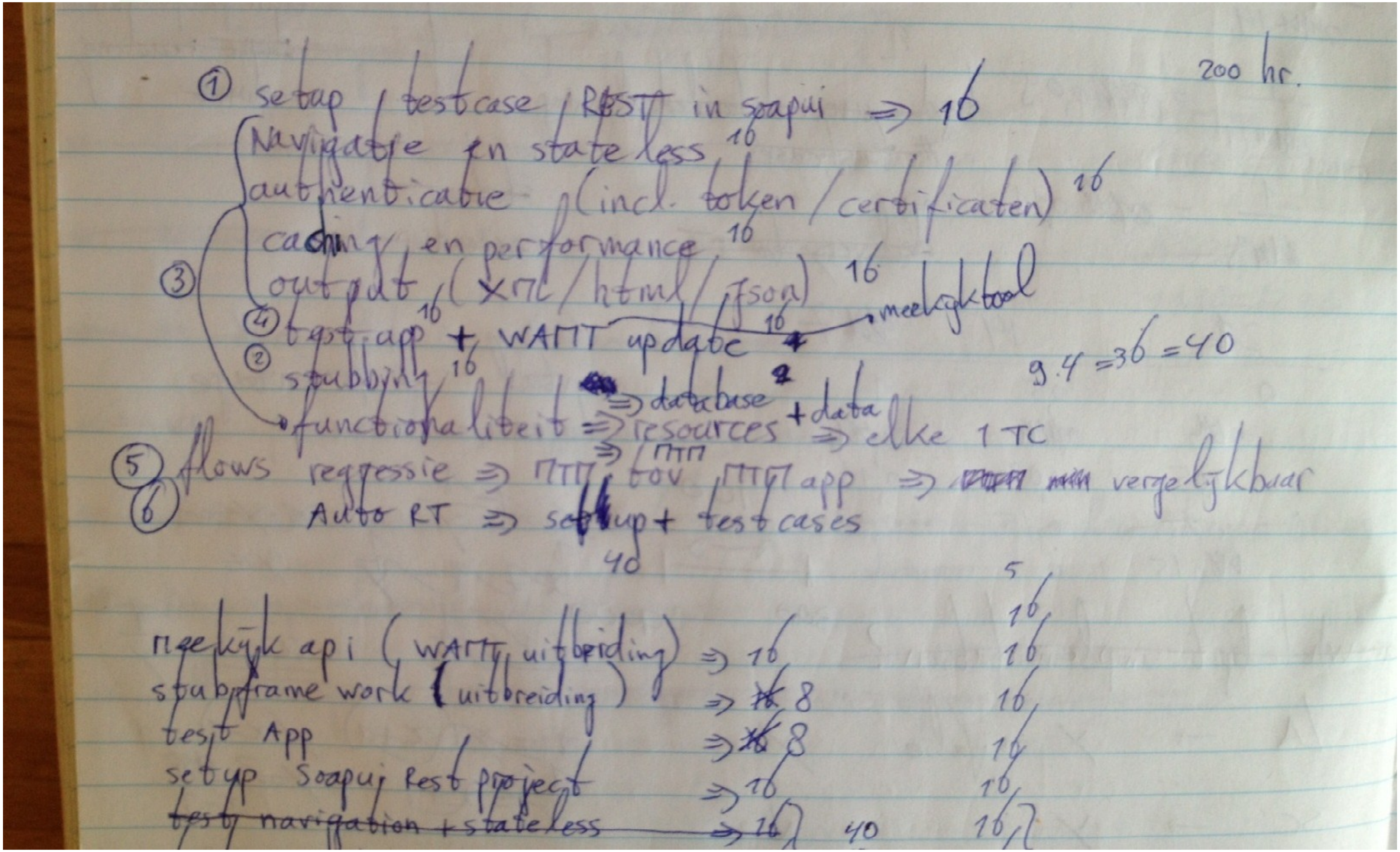


Resterende belminuten voor deze maand:	267:24 min
MMS	
Resterende SMS-berichten voor deze maand:	22 mms
i-200	



Vraag je toegangscode aan voor de App!

How-to test an API? – start from scratch



200 hr.

① setup / testcase / REST in soapui ⇒ 16

Navigatie en stateless 16

authenticatie (incl. token / certificaten) 16

caching en performance 16

③ output (xml / html / json) 16

④ test app + WAMT update 16 16 4 → meekijktool

② stubbing 16 ⇒ database 2 + data

functionaliteit ⇒ resources ⇒ elke 1 TC

9.4 = 36 = 40

⑤ flows

⑥ regressie ⇒ MTT ⇒ tov MTT app ⇒ [illegible] vergelijkbaar

Auto RT ⇒ setup + testcases

40

5
16

meekijk api (WAMT uitbreiding) ⇒ 16 16

stubframework (uitbreiding) ⇒ ~~16~~ 8 16

test App ⇒ ~~16~~ 8 16

setup Soapui Rest project ⇒ 16 16

~~test navigation + stateless~~ ⇒ 16 } 40 16 }

How-to test an API? – Typical risks

A. Unknown integration

B. Big variation of customer data

C. No control on the chain

D. Load is unknown

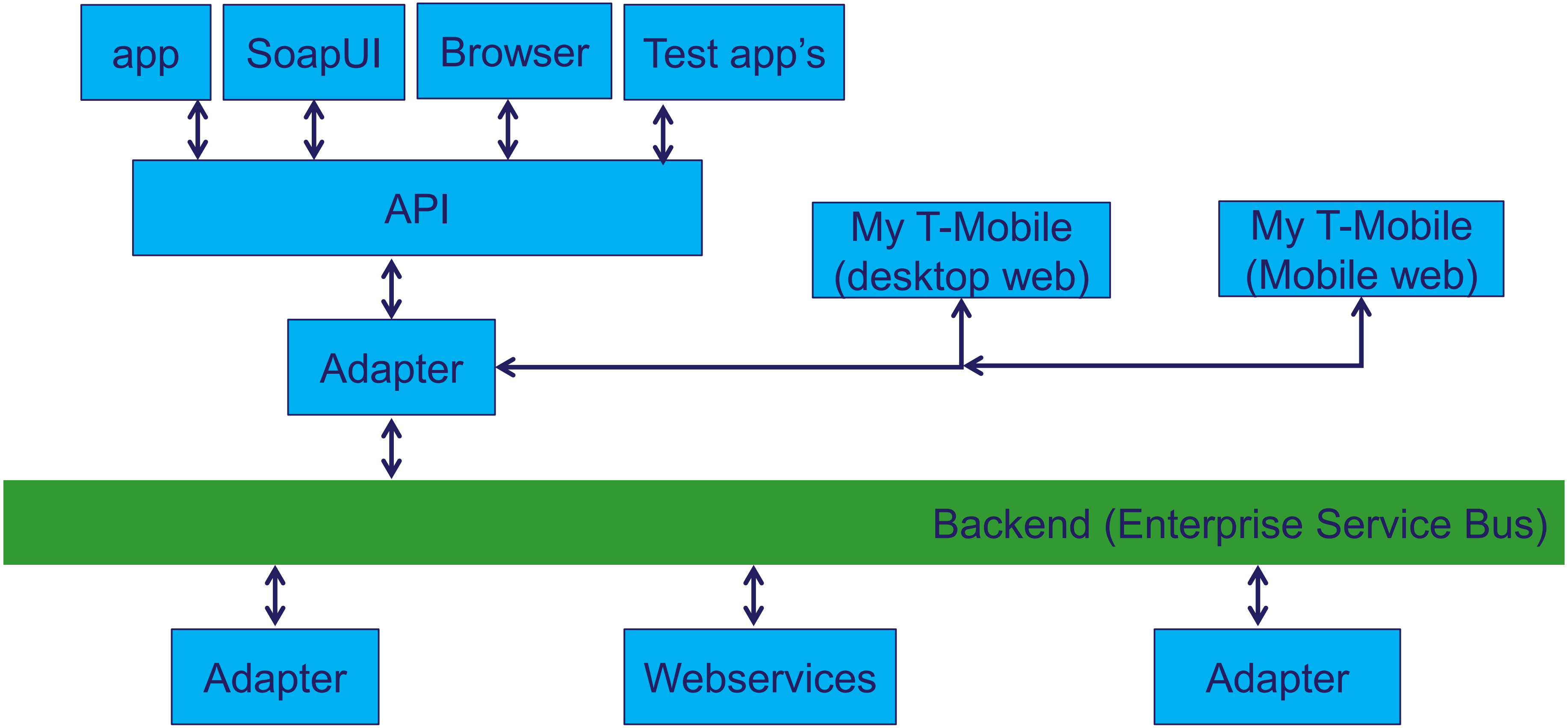
E. Wrong use of API

F. Dynamic scope

How-to test an API? – Strategy

- Early integration test with complete infrastructure
  - During development/system test system integration tests and Dogfooding (SoapUI, Browser, Windows taskbar app, Dev app)
  - Multiple integration phases
  - Prototype app (on Acceptance environment)

How-to test an API? – Test approach



Example defect – Cache key

- Test setup was: New session and clean all cookies/cache before each test case
- Each output for a resource is cached
- Defect was: it was cached with the same key for all customers
- Test scenario: login sequential with different users

Example defect – Error status

- Validation of HTTP-Statuses (Error and related status)
  - By using the general HTTP error code the app could not explicitly check what went wrong

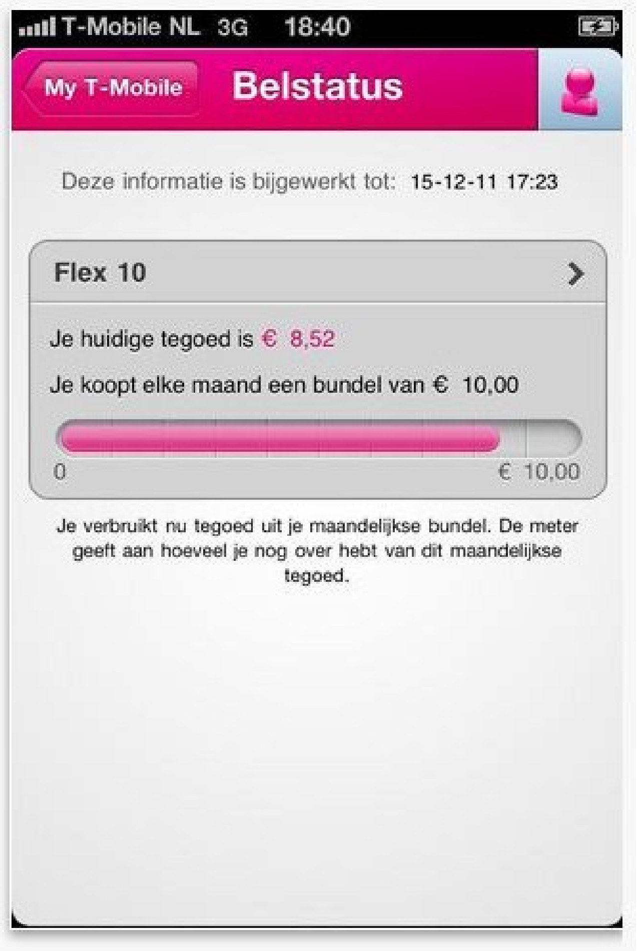
Error Code	Error Code Api	Status Description	Reason	Recommendation
401 - Unauthorized	1102	Invalid username or token.	The application supplied invalid credentials.	GUI: Ask for a different set of credentials.
401 - Unauthorized	1103	Account has been temporarily locked out.	Too many invalid attempts have been tried for these credentials, so the account is locked out for at least 1 minute	GUI: Ask the user to wait for at least 1 minute before retrying.

Example defect – Max. units

- Test data with big variation in rateplans
  - Total remaining bundle is determined by max. units.

```
<Bundle>
    <Name>Flex 10</Name>
    <FreeUnits xsi:nil="true" />
    <Unit>EURCENT</Unit>
    <MaxUnits>10</MaxUnits>
    <MaxUnitsPresentation>€ 10,00</MaxUnitsPresentation>
     <UsageAmount>148</UsageAmount>
    <UsageAmountPresentation>€ 1,48</UsageAmountPresentation>
</Bundle>

 <Bundle>
     <Name>Bel&SMS 15</Name>
     <FreeUnits xsi:nil="true" />
     <Unit>MIN/SMS</Unit>
     <MaxUnits>60/25</MaxUnits>
     <MaxUnitsPresentation>60 min./ 25 SMS</MaxUnitsPresentation>
     <UsageAmount>10/15</UsageAmount>
     <UsageAmountPresentation>10 min. / 15
      SMS</UsageAmountPresentation>
 </Bundle>
```



T-Mobile NL 3G 18:40

My T-Mobile

Belstatus

Deze informatie is bijgewerkt tot: 15-12-11 17:23

Flex 10

Je huidige tegoed is € 8,52

Je koopt elke maand een bundel van € 10,00

0 € 10,00

Je verbruikt nu tegoed uit je maandelijkse bundel. De meter geeft aan hoeveel je nog over hebt van dit maandelijkse tegoed.

Example defect – Menu name error

- Loyalty Programs: T-Mobile Extra or T-Mobile Speciaal
- Generic service only menu name different

- Interface design:
  - loyaltyProgramDescription value => TM Extra or TM Special

- Implementation T-Mobile frontend:

  If loyaltyProgramCode == "TME" then ProgramDescription = "T-Mobile Extra"

  If loyaltyProgramCode == "TMS" then ProgramDescription = "T-Mobile Speciaal"

Lessons learned - API

A. New test type: production tests
B. Command line
C. Scope
D. Security
E. No backup tricks

Lessons learned – API/app communication

A. T-Mobile is (still) seen as responsible for app while they only maintain API behind it

B. Presentation errors of data will always occur

C. Testing API with app can be very tempting to fix defects in app instead of API

D. More secure/controlled API needs more explanation

Lessons learned – Testing

A. Normal test techniques are usable
B. Defect analysis of chains: from backend to app
C. Run the automated regression test on API
D. New responsibilities: system testers have the most knowledge to give support in all phases
E. Experience with SOA/Interfaces/HTTP protocol/Tools helps
F. Experienced team with building interfaces (portals)

Summary

- Project
  - API and app are live, API enables a new world
  - API is now used by 250.000 users
  - Prepared for the future: Facebook, Windows, Blackberry apps are coming

- System testing
  - New responsibilities: support during all phases and production tests
  - API tests are early integration tests in a broader scope
  - Experienced team with SOA/interfaces and tools is a big plus

Questions or remarks?

Test possiblity of API at the Polteq stand

References

Source	Link
[Apple, 2009]	iPhone 3g Commercial "There's An App For That" http://www.youtube.com/watch?v=szrsfeyLzyg
[T-Mobile, 2011]	Customer API T-Mobile: https://capi.t-mobile.nl/
[Wiki,2012]	API : http://en.wikipedia.org/wiki/Application_programming_interface#Use_of_APIs_to_share_content REST architecture: http://en.wikipedia.org/wiki/Representational_state_transfer Internet Media Types: http://en.wikipedia.org/wiki/Internet_media_type#Type_application SOA: http://en.wikipedia.org/wiki/Service-oriented_architecture Sequence diagrams: http://nl.wikipedia.org/wiki/Unified_Modeling_Language Dogfooding: http://en.wikipedia.org/wiki/Eating_your_own_dog_food

References, *continued*

Source	Link
[SmartBear Software, 2011]	SoapUI with REST: http://www.soapui.org/REST-Testing/getting-started.html
[Tweakers, 2012]	More secure public API from NS: http://tweakers.net/nieuws/81389/ns-wil-apps-die-prijs-treinreis-tonen-vooraf-keuren.html
[D-Centralize, 2012]	Beltegoed Tulp: http://www.d-centralize.nl/beltegoed
[Belstatus, 2012]	Belstatus: http://www.p-edge.nl/belstatus
[Belstand, 2012]	Belstand: http://www.belstand.nl/
[Skindustries, 2012]	Bundels: http://www.skindustries.net/?p=portfolio&item=bundels

More information

- Marc van 't Veer
  Test consultant

  Polteq Dordrecht

- +31 (0) 6 46 63 61 48 (mob)
  http://www.polteq.com
  marc.vantveer@polteq.com

Polteq ondersteunt haar klanten:

- bij het uitvoeren en aansturen van testprojecten
- bij de inrichting en optimalisatie van testprocessen
- met praktijktrainingen en certificatieopleidingen

Polteq is gevestigd in:

- Amersfoort
- Amsterdam
- Dordrecht
- Groningen

Backup slides

How-to test an API? – Example test cases

Phase	Structure	Element	Test case	Defect
0. Setup	**Test application**	Setup	- Stub frame work - Tooling (SoapUI, browser, + add-ins, Fiddler, Altova XML Spy, FileZilla) - Interfaces and test data	
1. Development	Framework	Dogfooding	Check easy of use framework and how good it functions	
2. System testing	Functionality with and without stubs	• With and without stubs • Test applications	URL Structure, parameters • Header request : character set, media types, Verbs (GET/POST/UPDATE), HTTP(S) • Logging: Support defect analysis (server and database) • HTTP-statussen: Error and related status Common validation/Authentication/Authorization • Access codes (tokes) • Profiles (subscription active, SIM active, token status) • Customer/contract types • Caching: Duration, authorization, clearing, content, multiple users • Navigation: between resources Resources • Profiles: Business rules for each resource with variation customer data Output • Presentation forms and versions (supported not supported) Monitoring • Reporting and performance (load filter, backend availability Documentation	
	SIT	No stubs Integration with Tibco	Variation in customers, rateplans, prices, contract types, history, options with no stubs	
3. Integration testing	**Internal**	API with complete backend	No regression impact of other projects	
4. Acceptance testing	**External integration**	**API with prototype**	**First integration of app with API**	**Prototype app integration during acceptance testing:** **- Resource name of loyalty** **- Caching multiple users** **- HTTP statussen**
5. Production testing	**Real customers**	**API with live app**	**Real customers with app on production API**	**Production integration testing:** **- Variation in customers, rateplans, prices, contract types, history, options**
6. Regression	**Regression**	Reruns	- Manual and Automated regression tests	