delegate

DG – Theory into practice
2016-04-08 @ Delegate A/S HQ

Danmarks Tekniske Universitet DTU

IT University of Copenhagen

Agenda

- Matching of expectations
- Short introduction: Speaker and Delegate A/S
- Theory into practice
  - Delegate A/S palette of technologies (what we do)
    - Azure
    - CRM and SharePoint
  - Specific solutions where we have applied theory into practice
    - Timereg App
    - Data migration Dependency
    - SwaggerProvider and non-blocking Console Logger
    - XrmDefinitelyTyped and XrmContext
  - Open Source
- Summary
- Q&A

Matching of expectations

- What are your expectations for this talk?

Matching of expectations

- We want to show that it is possible to use what you are learning at your studies. You might have heard that some companies say to people coming from academia: “Now forget everything you have learned at University, now you are going learn something useful for the ****real life**** so you can be a specialist in some random Framework”

- Therefore, we will showcase the technologies we use, which are built on computer science foundations (several paradigms) and showcase a few examples that students like you have made for us.

Note: Feel free to interrupt and ask questions

Intro – About me (very shortly)

- Ramón Soto Mathiesen
- MSc. Computer Science from DIKU, with Minors in Mathematics (2005 – 10)
- Managing Specialist |> CTO of CRM Department @ Delegate A/S
  - ER-modeling, WSDL, OData (REST API)
- F# / C# / JavaScript / C++: Delegate A/S @ GitHub
- Blog: http://blog.stermon.com/
- Recent Talks:
  - 2016-02-08: CRMUG – MS CRM Solution Packager
  - 2016-03-12: SPBG – A combination of MS SharePoint & CRM to ensure atomic transactions

Intro – Delegate A/S

- IT-Consultancy known for their SharePoint solutions and established in 2006
- Expanded in 2013 with CRM and Office 365 departments
- Last year we formed a Windows Azure (Cloud) department (Simon)
- We focus on technique and are "**proud**" to call ourselves "**nerds**"
- The focus is on employees with career plan, training and social events
- We are located both in Aarhus (11) and Copenhagen (49)
- The company's motto: "**We must be the best, not necessarily the biggest**"

GREAT PLACE TO WORK IT-Branchens Bedste Arbejdspladser 2013 Denmark

GREAT PLACE TO WORK® IT-Branchens Bedste Mindre Arbejdspladser 2014 Denmark

GREAT PLACE TO WORK® Danmarks Bedste Mindre Arbejdspladser 2014 Denmark

GREAT PLACE TO WORK® Danmarks Bedste Arbejdspladser® 2015 Denmark

Microsoft Partner Award 2014

Årets CRM Partner

Microsoft

Intro – Delegate A/S

UM (Udenrigsministeriet) best SP intranet 2015

- The company's motto: "**We must be the best, not necessarily the biggest**"

GREAT PLACE TO WORK IT-Branchens Bedste Arbejdspladser 2013 Denmark

GREAT PLACE TO WORK® IT-Branchens Bedste Mindre Arbejdspladser 2014 Denmark

GREAT PLACE TO WORK® Danmarks Bedste Mindre Arbejdspladser 2014 Denmark

GREAT PLACE TO WORK® Danmarks Bedste Arbejdspladser® 2015 Denmark

Microsoft Partner Award 2014

Årets CRM Partner

Microsoft

Palette of technologies (everything under the Office365 umbrella)

Microsoft Azure

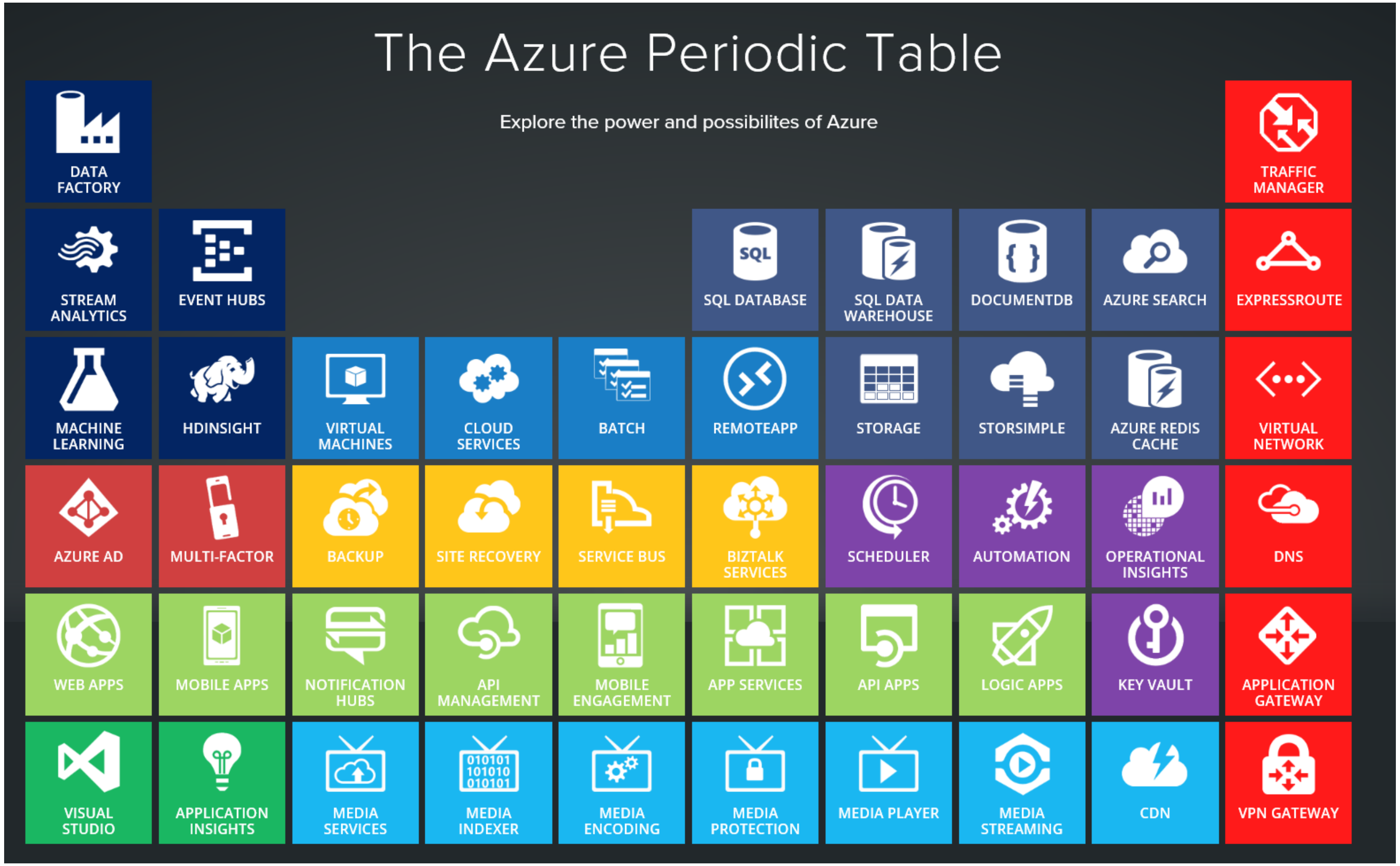
Microsoft Dynamics

SharePoint

Azure

CRM and SharePoint

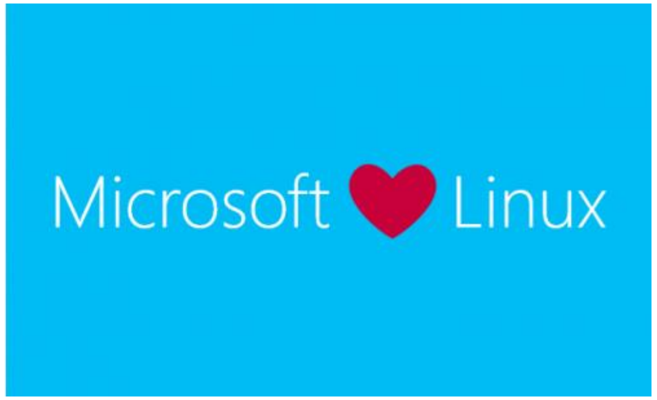
Azure (Cloud technologies)



Azure (Cloud technologies)

- Just think of Azure as a place where you have a lot of computer power that you can rent, for a decent amount of money, and be able to make some task that you couldn't afford if you had to setup the infrastructure yourself.

- You can pretty much do anything in Azure (yes, even Linux)



- Before in the "good old days" we weren't allowed to code from scratch, due to licensing on Framework products, but now, as long as the code is placed on the Azure Cloud. It's all good ☺ As it is right now, Microsoft don't really care what you do as long as your ***for-loops*** run on their Cloud service (and not on Amazon's/Google/…)

CRM and SharePoint (Frameworks)

- Even though it’s fun to code from scratch (helps the creative process) you really don’t want to re-invent the wheel every time you start a new project
- Therefore it’s easier to make new projects based on a finished product which can subsequently be extended to suit a specific customer
- And it fits quite nicely with Delegates Agile (Scrum) culture

CRM and SharePoint (Frameworks)

SharePoint	Unstructured & Semi-structured Data	Social Engagement & Collaboration	Information Evolution & Storage	Informal Rules	Enterprise Search	Across Teams & Domains
CRM	Structured Relational Data	Interactions, Activities & Tasks	Information Generation & Analytics	Formalized Processes	Data Query & Filter	Within Teams & Domains

Comparison between the two frameworks

CRM and SharePoint (Frameworks)

- SharePoint
  - Built on top of a relational database, which is abstracted completely away from end-users as every data entity is represented as a list
  - Lists can point to other lists (lookup fields, think of it as foreign keys but logic is done in the application- and not the data- layer) and are easily expandable by end-users
  - To put some control, we provide type-safety to generated C# classes based on IOC (Inversion of Control, Frameworks call custom code) that ensures that list created from source-control, comply with business logic (as long as list just get addition of properties). For the client-code we use TypeScript to give an OO approach and safetyness to development and deploy

- CRM
  - Built on top of a relational database as well where business analyst expand it with new entities from an administration module. This gives real foreign key constraints and therefore we have atomic transactions (ACID)
  - Less end-user friendly as every change to the system has to go through DEV → TEST → PROD. We help our customers to make this process smoothly (DAXIF#) so they don’t have to depend on us
  - We don’t use IOC (custom code calls into generic libraries) and we tend to loose a bit up (late-bound) when we code applications that can work on several CRM solutions (we can’t see into the future on how customers will model their domains). We use TypeScript for the client-side as well

Specific solutions where we have applied theory into practice

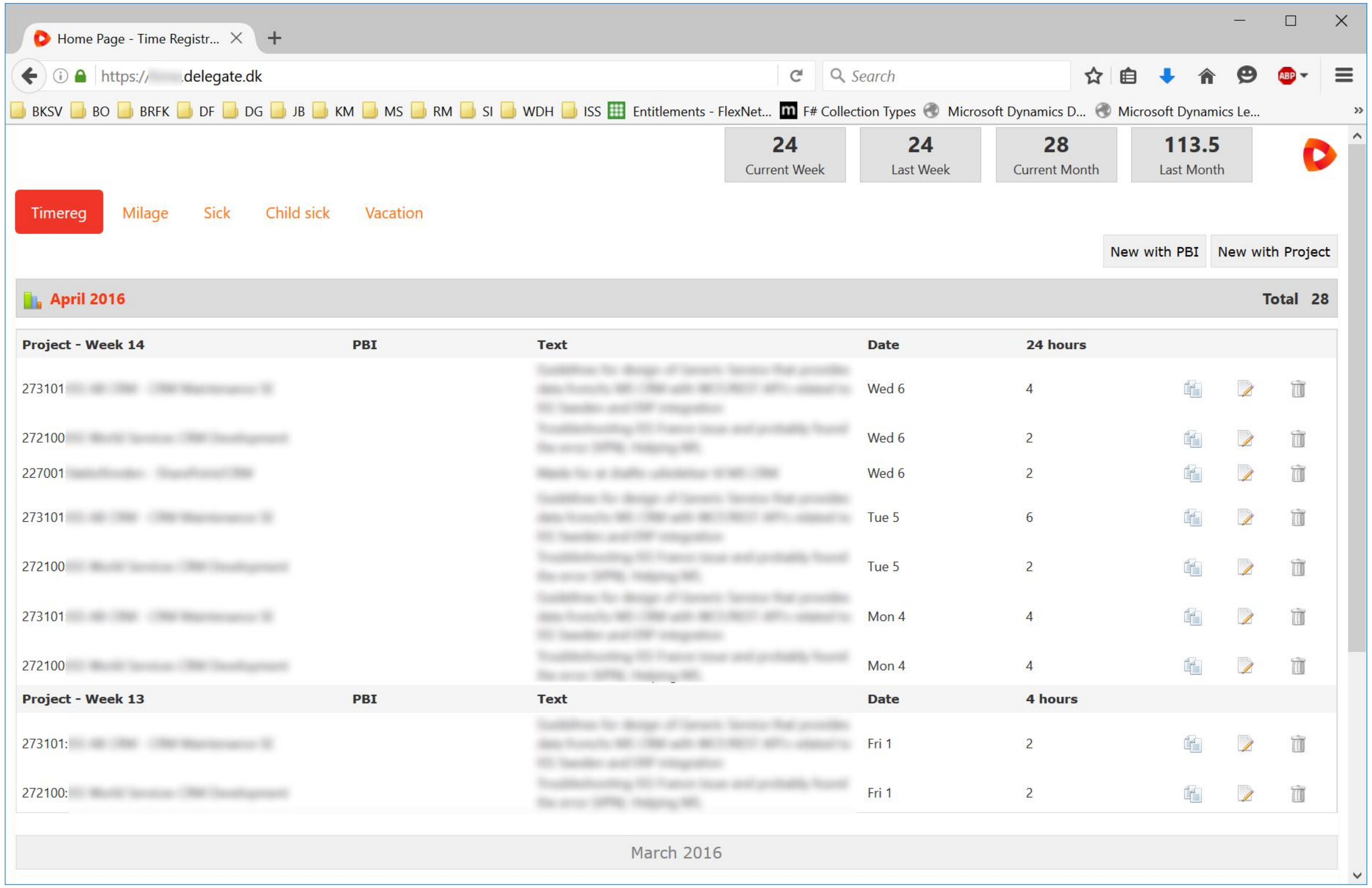
“which we can talk about”

Note: Slides will be made publically available, we can’t showcase specific customer cases ☺

Timereg App – Alexander (SharePoint and Tiimo)

- Alexander:
  - B.Sc. in Robotics SDU
  - M.Sc. in Engineering DTU
  - Started as student (last year)
- Task:
  - Maintain and add new features to our Timereg App
- Proposed solution:
  - Applying design pattern (MVVM) as well as replacing client side code with Angular and TypeScript to ensure less development errors

Timereg App – Alexander (SharePoint and Tiimo)



24	24	28	113.5
Current Week	Last Week	Current Month	Last Month

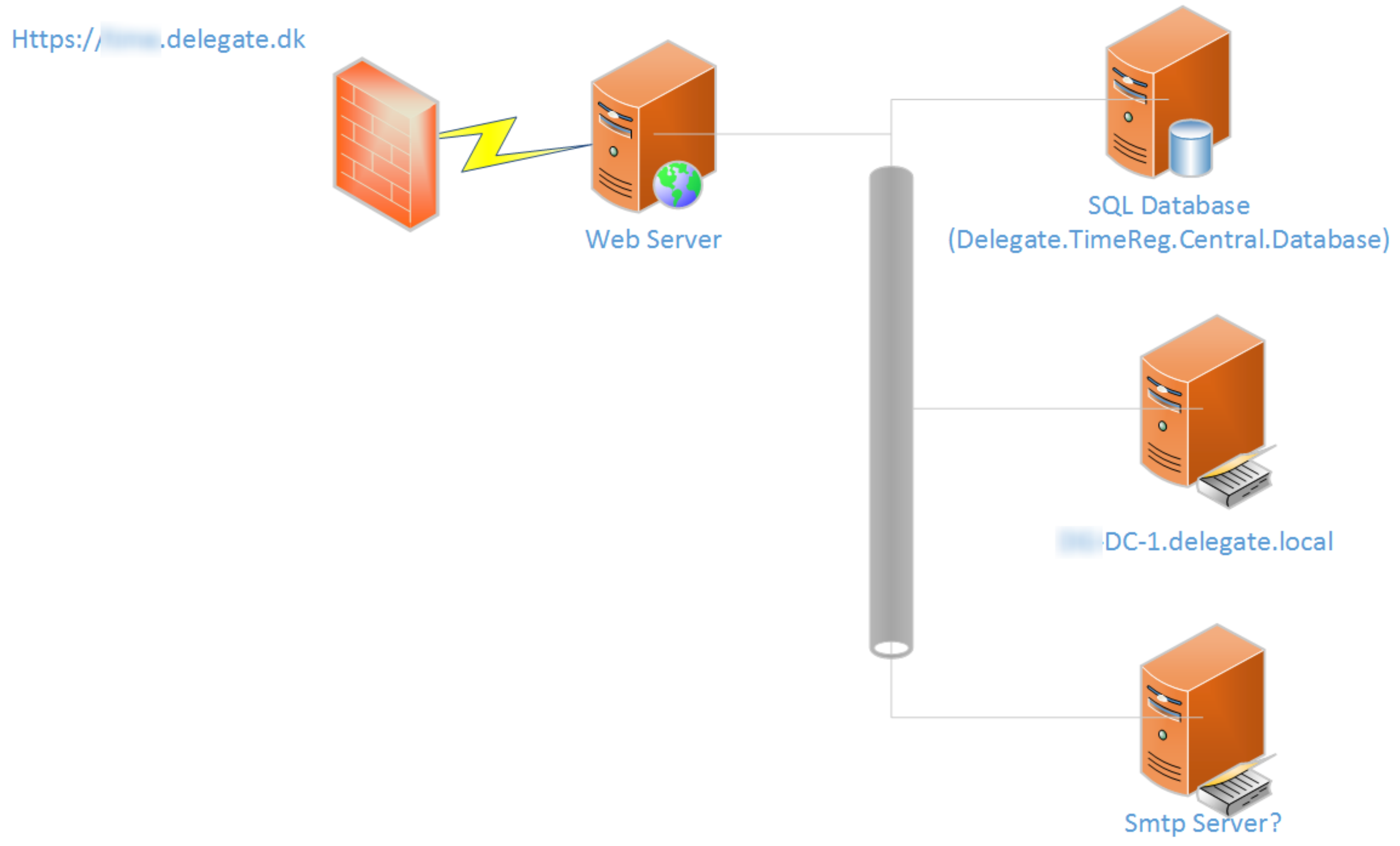
April 2016 — **Total 28**

Project - Week 14	PBI	Text	Date	24 hours
273101 [redacted]		[redacted]	Wed 6	4
272100 [redacted]		[redacted]	Wed 6	2
227001 [redacted]		[redacted]	Wed 6	2
273101 [redacted]		[redacted]	Tue 5	6
272100 [redacted]		[redacted]	Tue 5	2
273101 [redacted]		[redacted]	Mon 4	4
272100 [redacted]		[redacted]	Mon 4	4
Project - Week 13	**PBI**	**Text**	**Date**	**4 hours**
273101: [redacted]		[redacted]	Fri 1	2
272100: [redacted]		[redacted]	Fri 1	2

March 2016

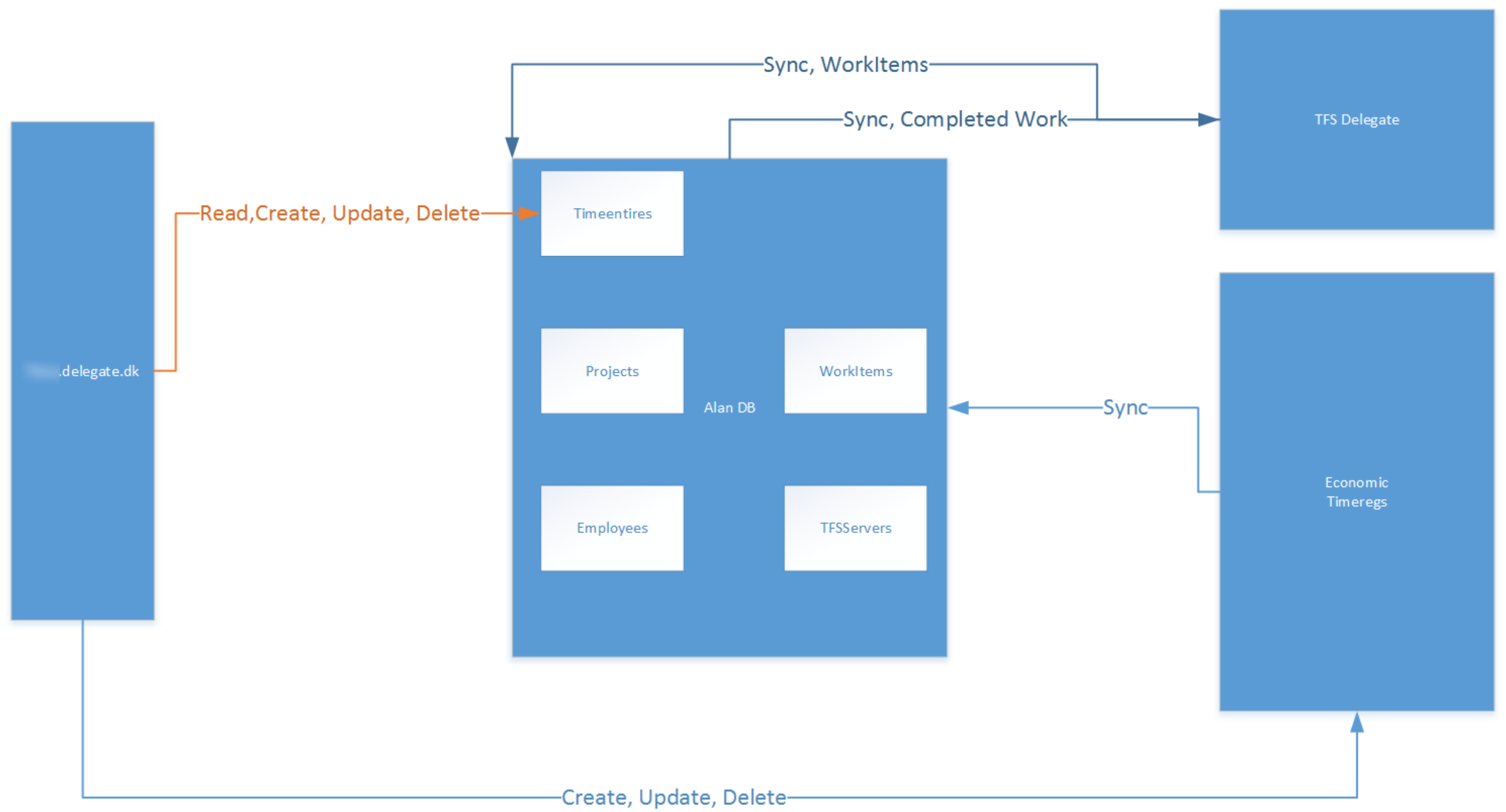
Web User Interface

Timereg App – Alexander (SharePoint and Tiimo)


Https://[redacted].delegate.dk
Web Server
SQL Database
(Delegate.TimeReg.Central.Database)
[redacted]DC-1.delegate.local
Smtp Server?

Infrastructure

Timereg App – Alexander (SharePoint and Tiimo)



CRUD flow diagram

Data migration Dependency – Niclas aka Dr. Yammer (CRM)

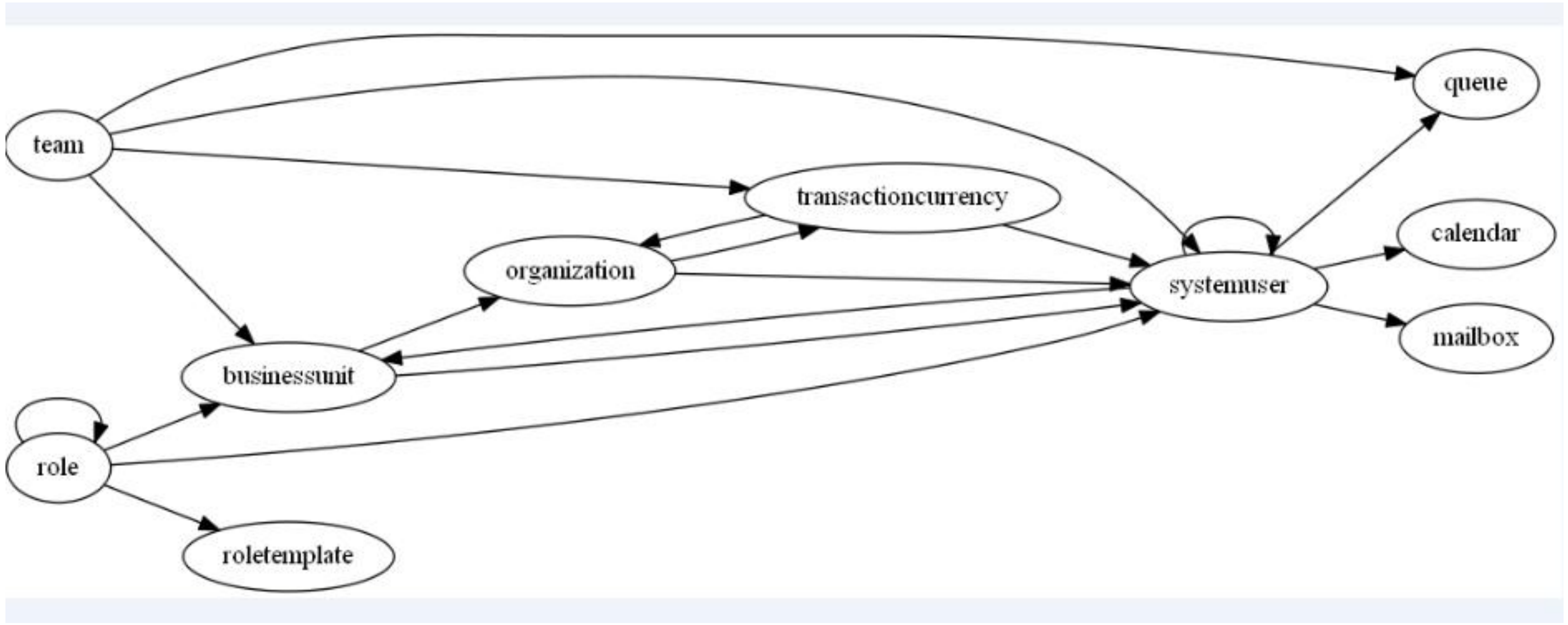
- Niclas aka Dr. Yammer
  - B.Sc. in Software Engineering ITU
  - Former student employee

- Task:
  - Ensure we use the least amount of time migrating data

- Proposed solution:
  - As most data models have relations (relational databases), when you represent them as a graph, you will discover that there are a lot of dependencies that will make it more complex to transfer data in parallel due to data pointing to entities that are not created. This issue is solved by transforming the graph representing the data model to a DAG (Directed acyclic graph)

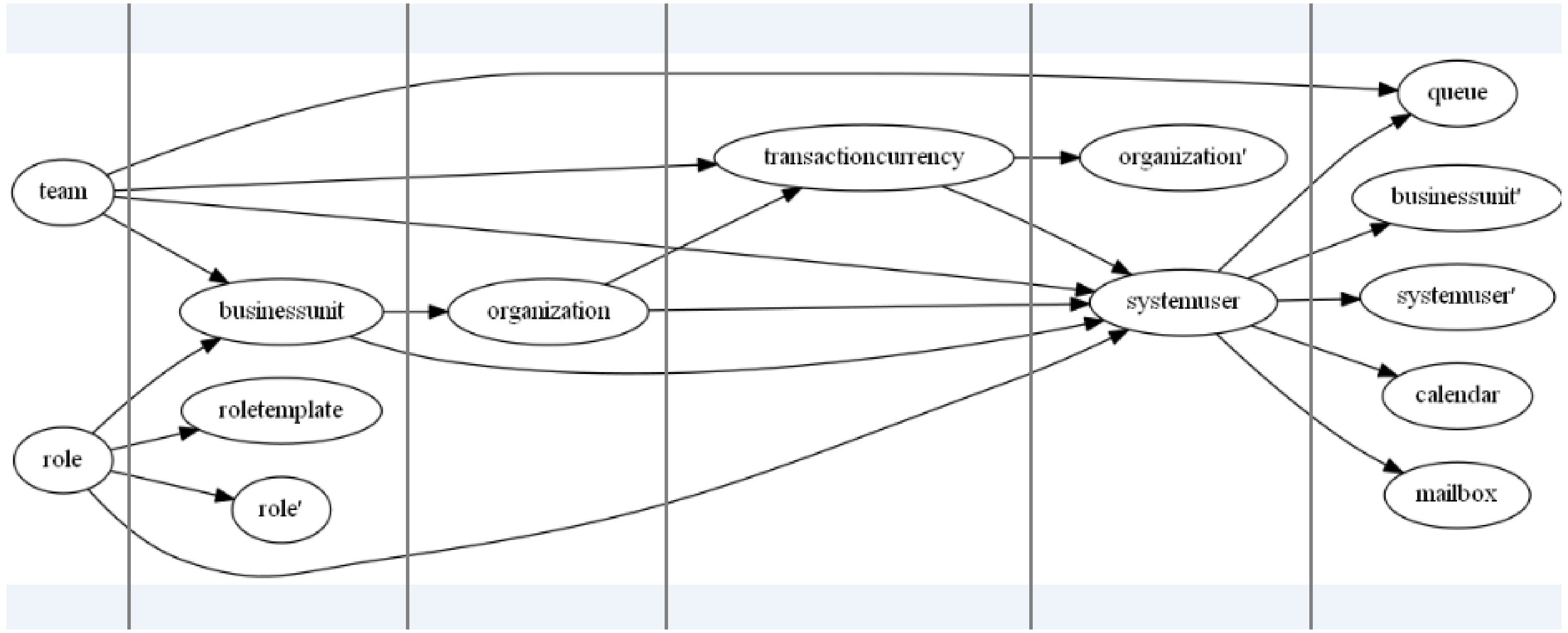
Note: Asymptotically, 2N is still linear but, N + ½N, is way faster.

Data migration Dependency – Niclas aka Dr. Yammer (CRM)



Graph representing data model with a lot of dependencies (also self referring)

Data migration Dependency – Niclas aka Dr. Yammer (CRM)

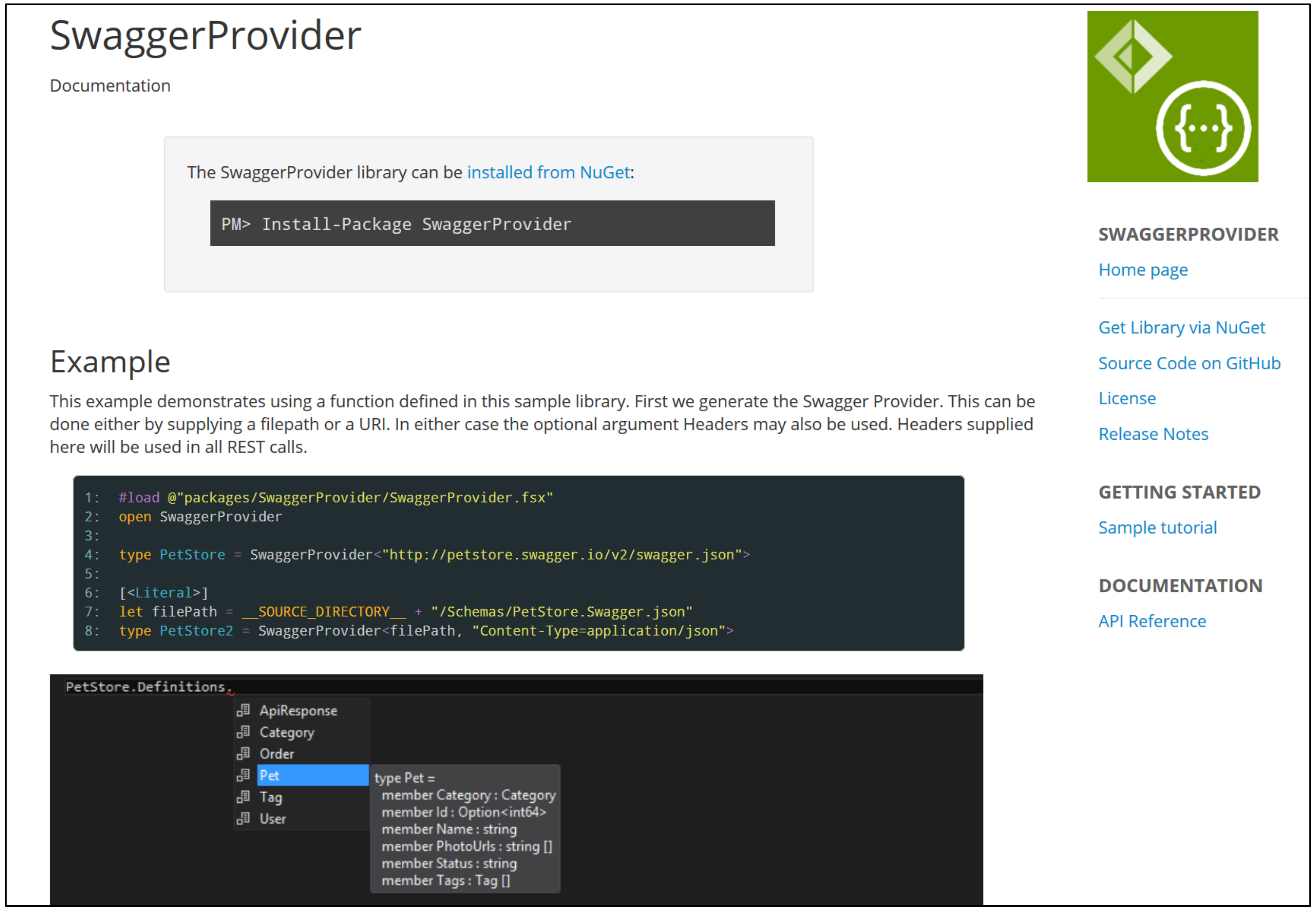


Graph converted to a DAG (Directed acyclic graph) without no dependencies allowing to execute data loads in each segment in parallel

SwaggerProvider and non-blocking Console Logger – Tobias (Azure and CRM)

- Tobias:
  - B.Sc. in Engineering DTU
  - Still a student (you might have seen him around) ☺
- Task(s):
  - Tool to test our SharePoint site provisioning engine
  - Replace blocking Console Logger from DAXIF#
- Proposed solution:
  - Contribute to SwaggerProvider (Open Source) project. Side-effect of this work was that code was re-used for another project: Delegate.Office365.Common.WorkflowActions (Brandon)
  - The previous Console Logger was based on .NET Console.WriteLine, which is blocking. The approach was to introduce an Agent (reducer) that would receive messages from DAXIF# processes

SwaggerProvider and non-blocking Console Logger – Tobias (Azure and CRM)



SwaggerProvider

Documentation

The SwaggerProvider library can be installed from NuGet:

```
PM> Install-Package SwaggerProvider
```

Example

This example demonstrates using a function defined in this sample library. First we generate the Swagger Provider. This can be done either by supplying a filepath or a URI. In either case the optional argument Headers may also be used. Headers supplied here will be used in all REST calls.

```
#load @"packages/SwaggerProvider/SwaggerProvider.fsx"
open SwaggerProvider

type PetStore = SwaggerProvider<"http://petstore.swagger.io/v2/swagger.json">

[<Literal>]
let filePath = __SOURCE_DIRECTORY__ + "/Schemas/PetStore.Swagger.json"
type PetStore2 = SwaggerProvider<filePath, "Content-Type=application/json">
```

```
PetStore.Definitions.
```

- ApiResponse
- Category
- Order
- Pet
- Tag
- User

```
type Pet =
  member Category : Category
  member Id : Option<int64>
  member Name : string
  member PhotoUrls : string []
  member Status : string
  member Tags : Tag []
```

SWAGGERPROVIDER

Home page

Get Library via NuGet

Source Code on GitHub

License

Release Notes

GETTING STARTED

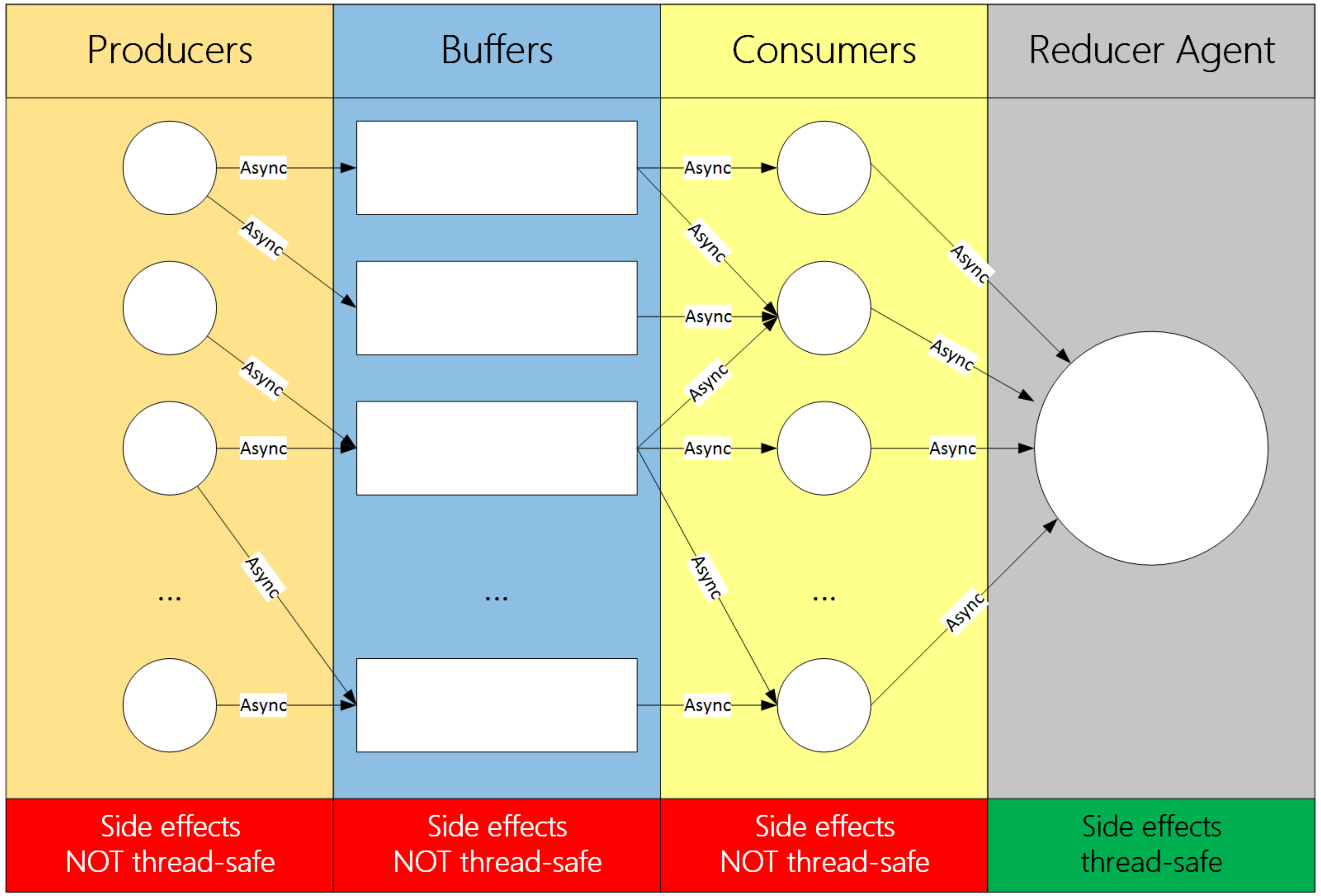
Sample tutorial

DOCUMENTATION

API Reference

TypeProviders provide "erased-types" on the fly from your IDE

SwaggerProvider and non-blocking Console Logger – Tobias (Azure and CRM)


Producers
Buffers
Consumers
Reducer Agent
Async
...
Side effects NOT thread-safe
Side effects NOT thread-safe
Side effects NOT thread-safe
Side effects thread-safe

A single reducer agent that will print asynchronously to the console

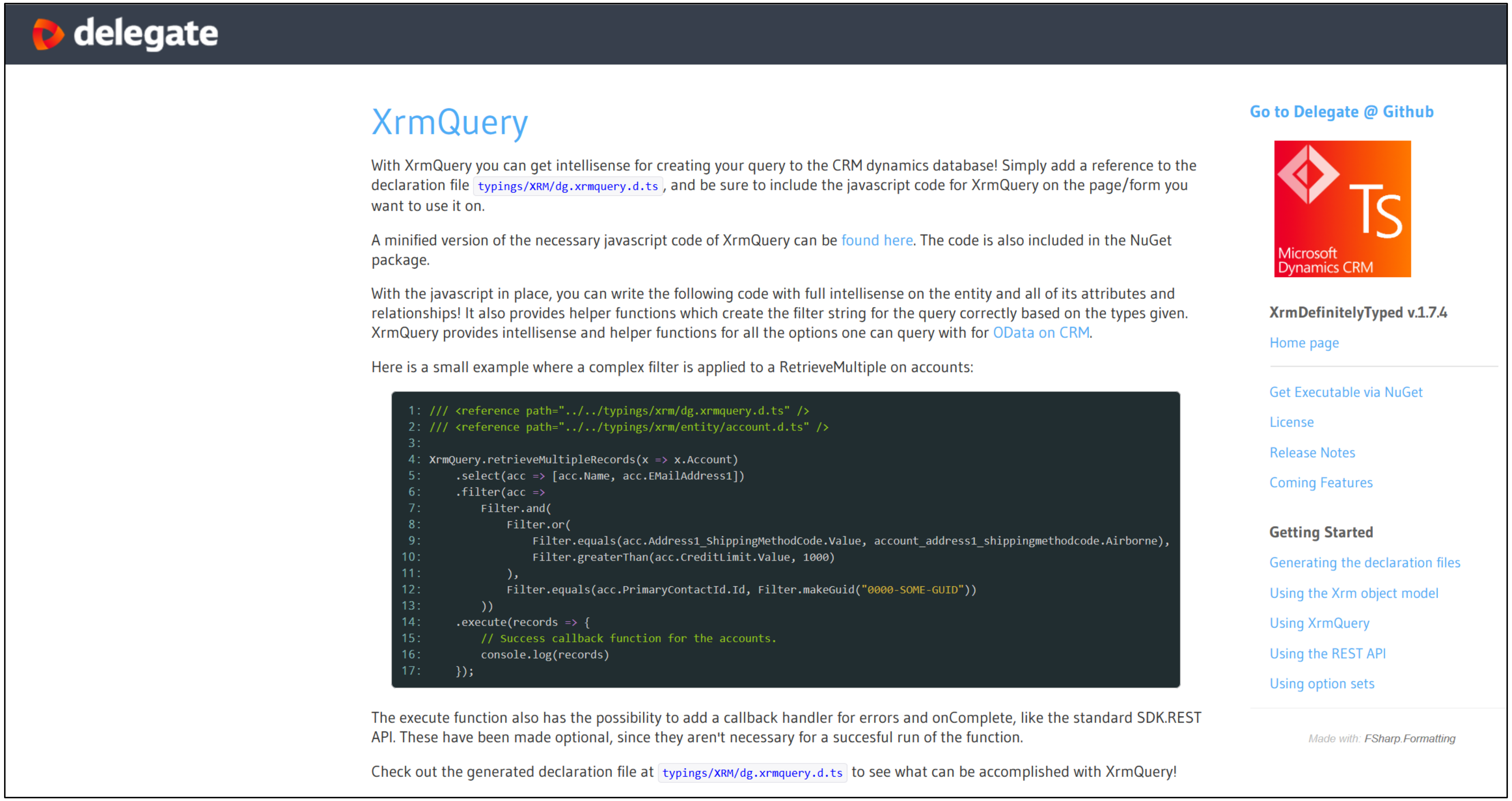
XrmDefinitelyTyped and XrmContext – Martin aka Mr. T (CRM)

- Martin aka Mr. T
  - M.Sc. in Engineering DTU
  - 6 months at Facebook in Silicon Valley
  - Started as student (two years ago)

- (self proposed) Task(s):
  - Lack of types in MS CRM client-side code
  - MS CRM created type-safe classes aren't that safe after all …
- Proposed solution(s):
  - XrmDefinitelyTyped: TypeScript Declaration File Generator for MS CRM
  - XrmContext: Tool to generate early-bound .NET classes for server-side CRM coding

Note: Both tools are *mandatory* for all our CRM projects

XrmDefinitelyTyped and XrmContext – Martin aka Mr. T (CRM)



delegate

XrmQuery

With XrmQuery you can get intellisense for creating your query to the CRM dynamics database! Simply add a reference to the declaration file `typings/XRM/dg.xrmquery.d.ts`, and be sure to include the javascript code for XrmQuery on the page/form you want to use it on.

A minified version of the necessary javascript code of XrmQuery can be found here. The code is also included in the NuGet package.

With the javascript in place, you can write the following code with full intellisense on the entity and all of its attributes and relationships! It also provides helper functions which create the filter string for the query correctly based on the types given. XrmQuery provides intellisense and helper functions for all the options one can query with for OData on CRM.

Here is a small example where a complex filter is applied to a RetrieveMultiple on accounts:

```
/// <reference path="../../typings/xrm/dg.xrmquery.d.ts" />
/// <reference path="../../typings/xrm/entity/account.d.ts" />

XrmQuery.retrieveMultipleRecords(x => x.Account)
    .select(acc => [acc.Name, acc.EMailAddress1])
    .filter(acc =>
        Filter.and(
            Filter.or(
                Filter.equals(acc.Address1_ShippingMethodCode.Value, account_address1_shippingmethodcode.Airborne),
                Filter.greaterThan(acc.CreditLimit.Value, 1000)
            ),
            Filter.equals(acc.PrimaryContactId.Id, Filter.makeGuid("0000-SOME-GUID"))
        ))
    .execute(records => {
        // Success callback function for the accounts.
        console.log(records)
    });
```

The execute function also has the possibility to add a callback handler for errors and onComplete, like the standard SDK.REST API. These have been made optional, since they aren't necessary for a succesful run of the function.

Check out the generated declaration file at `typings/XRM/dg.xrmquery.d.ts` to see what can be accomplished with XrmQuery!

Go to Delegate @ Github

XrmDefinitelyTyped v.1.7.4

Home page

Get Executable via NuGet

License

Release Notes

Coming Features

Getting Started

Generating the declaration files

Using the Xrm object model

Using XrmQuery

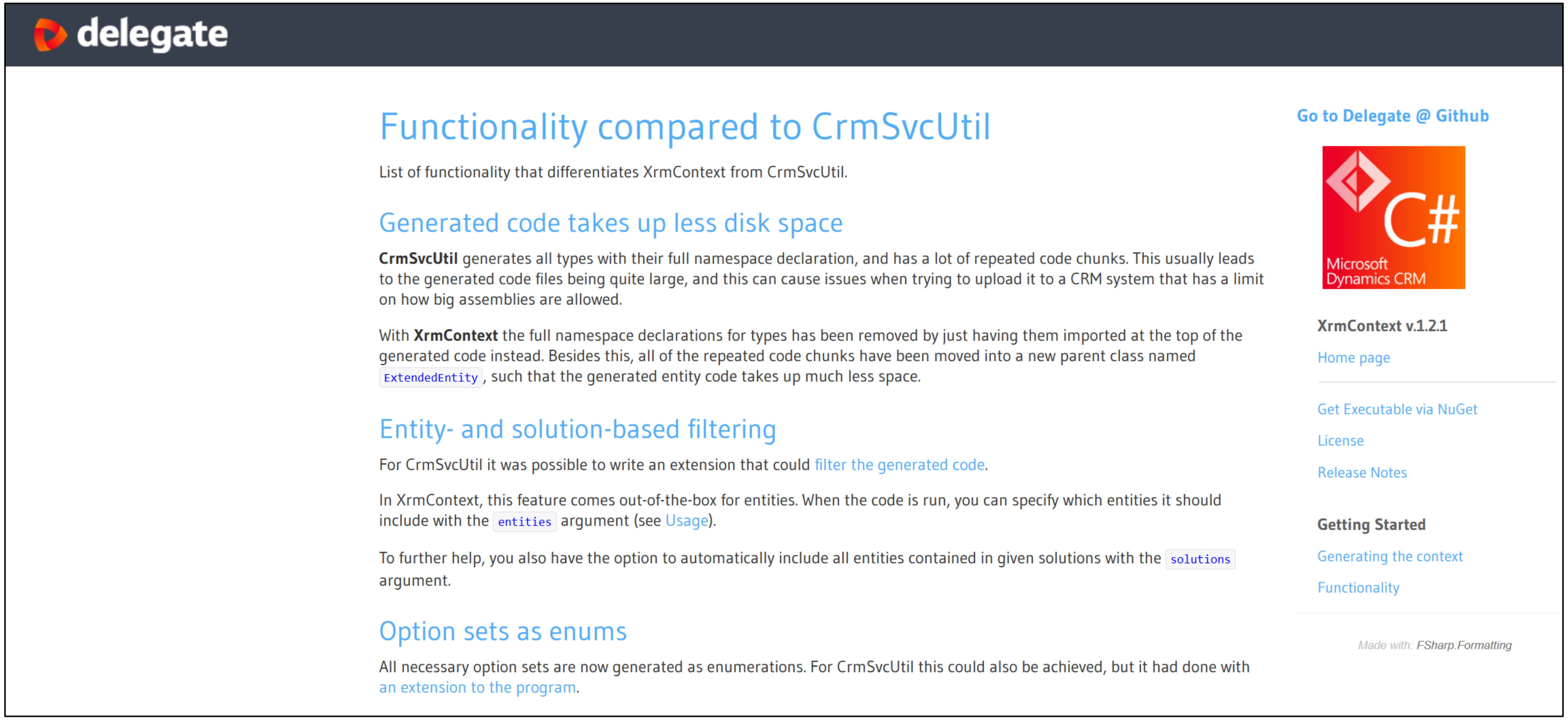
Using the REST API

Using option sets

Made with: FSharp.Formatting

C#'s LINQ alike code in TypeScript

XrmDefinitelyTyped and XrmContext – Martin aka Mr. T (CRM)


delegate

Functionality compared to CrmSvcUtil

List of functionality that differentiates XrmContext from CrmSvcUtil.

Generated code takes up less disk space

CrmSvcUtil generates all types with their full namespace declaration, and has a lot of repeated code chunks. This usually leads to the generated code files being quite large, and this can cause issues when trying to upload it to a CRM system that has a limit on how big assemblies are allowed.

With **XrmContext** the full namespace declarations for types has been removed by just having them imported at the top of the generated code instead. Besides this, all of the repeated code chunks have been moved into a new parent class named `ExtendedEntity`, such that the generated entity code takes up much less space.

Entity- and solution-based filtering

For CrmSvcUtil it was possible to write an extension that could filter the generated code.

In XrmContext, this feature comes out-of-the-box for entities. When the code is run, you can specify which entities it should include with the `entities` argument (see Usage).

To further help, you also have the option to automatically include all entities contained in given solutions with the `solutions` argument.

Option sets as enums

All necessary option sets are now generated as enumerations. For CrmSvcUtil this could also be achieved, but it had done with an extension to the program.

Go to Delegate @ Github

Microsoft Dynamics CRM

XrmContext v.1.2.1

Home page

Get Executable via NuGet

License

Release Notes

Getting Started

Generating the context

Functionality

Made with: FSharp.Formatting

Better generated code than the provided tool from MS ☺

Open Source

open source initiative®

We are just following in Microsoft's footsteps ☺

Open Source

delegate

@ GitHub

DAXIF#
Delegate Automated Xrm Installation Framework

Delegate.SPOcopy
SharePoint Online copy.

XrmDefinitelyTyped
TypeScript Declaration File Generator for MS CRM

LotusNotesDumper
Lotus Notes to MS CRM + SP
(Coming soon)

XrmContext
Tool to generate early-bound .NET classes for server-side CRM coding.

ƒ(x)RM
Functional Relationship Management
(Coming soon)

Delegate.Sandbox
I/O side-effects safe code by using a sandbox computation expression.

Open Source Tools @ GitHub: http://delegateas.github.io/

Summary

- We hope that you can see that even though we don’t make ***rocket science***, we are still doing some interesting projects where you are able to use a wide variety of the knowledge that you learn on a daily basis at University and most important, if you are able to convince us, you might change the way we do things ☺

- We haven’t shown customer projects as we might make these slides public but, in a bit you will have plenty of time to talk to other of our consultants “beer in hand” …

- … but first (last) …

Questions and answers

Q&A

Friday bar and Foosball tournament

Beer, snaks and "hygge"