

DWX 2013
Thorsten Hans

New SharePoint API's

about me

- Thorsten Hans
- SharePoint MVP @ Experts Inside
- th@expertsinside.com | th@dotnet-rocks.de
- <http://www.dotnet-rocks.de>
- <http://www.expertsinside.com>
- @ThorstenHans

about me

- Thorsten Hans
- SharePoint MVP @ Experts Inside
- th@expertsinside.com | th@dotnet-rocks.de
- <http://www.dotnet-rocks.de>
- <http://www.expertsinside.com>
- @ThorstenHans



Roadmap

REST API

When to use which API

JavaScript API (JSOM)

Demos

CLR API (CSOM)

Data & Services

SharePoint?!

SharePoint?!

Who is developing software for / on
SharePoint?

Defining SharePoint

Defining SharePoint

SharePoint is providing **data** and
services

API - History

SharePoint 2007

- Server-Side Object Model
- RPC calls (owssrv.dll)
- ASMX WebServices

API - History

SharePoint 2010

- Server-Side Object Model
- RPC calls (owssrv.dll)
- ASMX WebServices
- ListData.svc (REST'ish Service)
- Client-Side Object Model

SharePoint 2013

- Support for Server-Side Object Model (OnPremise)
- New REST API's (Data and Services)
- Extended Client Object Models (Data and Services)
- ASMX WebServices and RPC calls are **deprecated**

For Developers

By enabling SharePoint **Apps** and the new **API** set we're able to **interact** with
SharePoint **as platform!**

Our Apps won't run on (or in) SharePoint

For Developers

It's up to you

Server-Side API

For Developers

It's up to you

PS

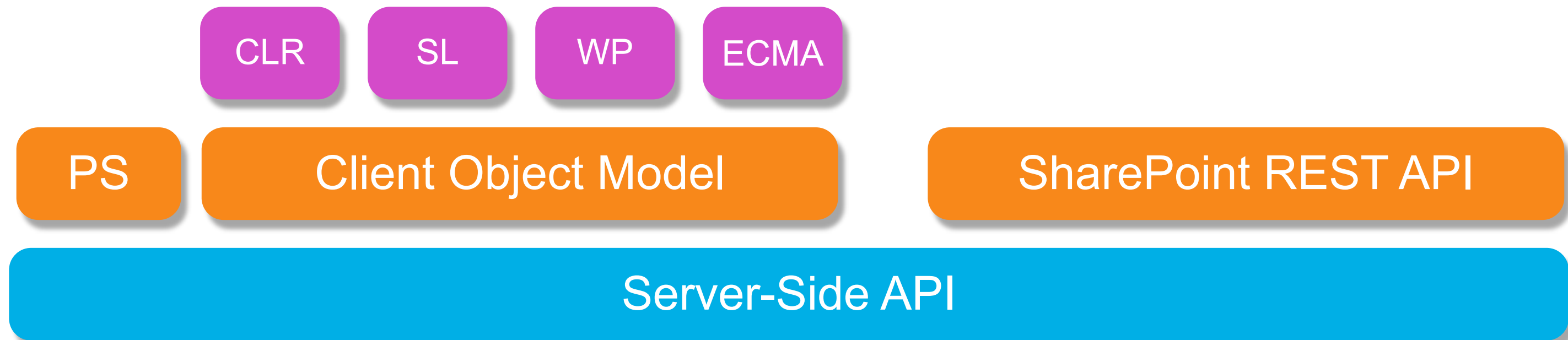
Client Object Model

SharePoint REST API

Server-Side API

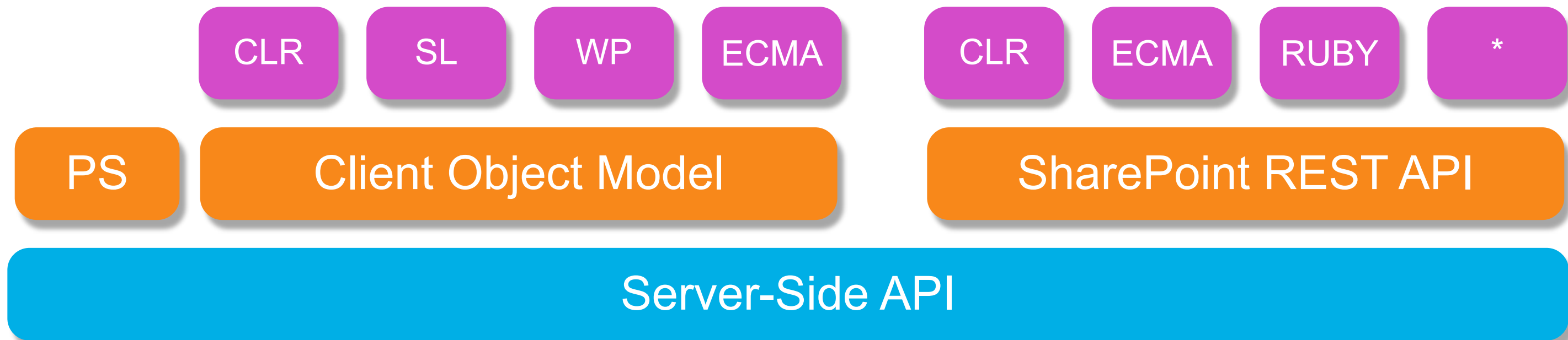
For Developers

It's up to you



For Developers

It's up to you



SharePoint Client Object Model

- Supported by
 - SharePoint OnPremise
 - SharePoint Online (Office 365)

Client Side Object Model

- Introduced with SharePoint 2010
- CRUD for Data
- Similar approach across technologies
- Request Batching 'n Trimming

JSOM

JavaScript Object Model (JSOM)
is one **Client Object Model** flavor.

JSOM

It's only supporting `async` calls!

JSOM

It's mapping CLR **properties**
to **getter**- and **setter**-functions

JSOM

It's mapping CLR **properties**
to **getter**- and **setter**-functions

```
title = web.Title;  
web.Title = "DWX Site";
```

JSOM

It's mapping CLR **properties**
to **getter**- and **setter**-functions

```
title = web.Title;  
web.Title = "DWX Site";
```

```
title = web.get_title();  
web.set_title("DWX Site")
```

Recap - Request Batching

Recap - Request Batching

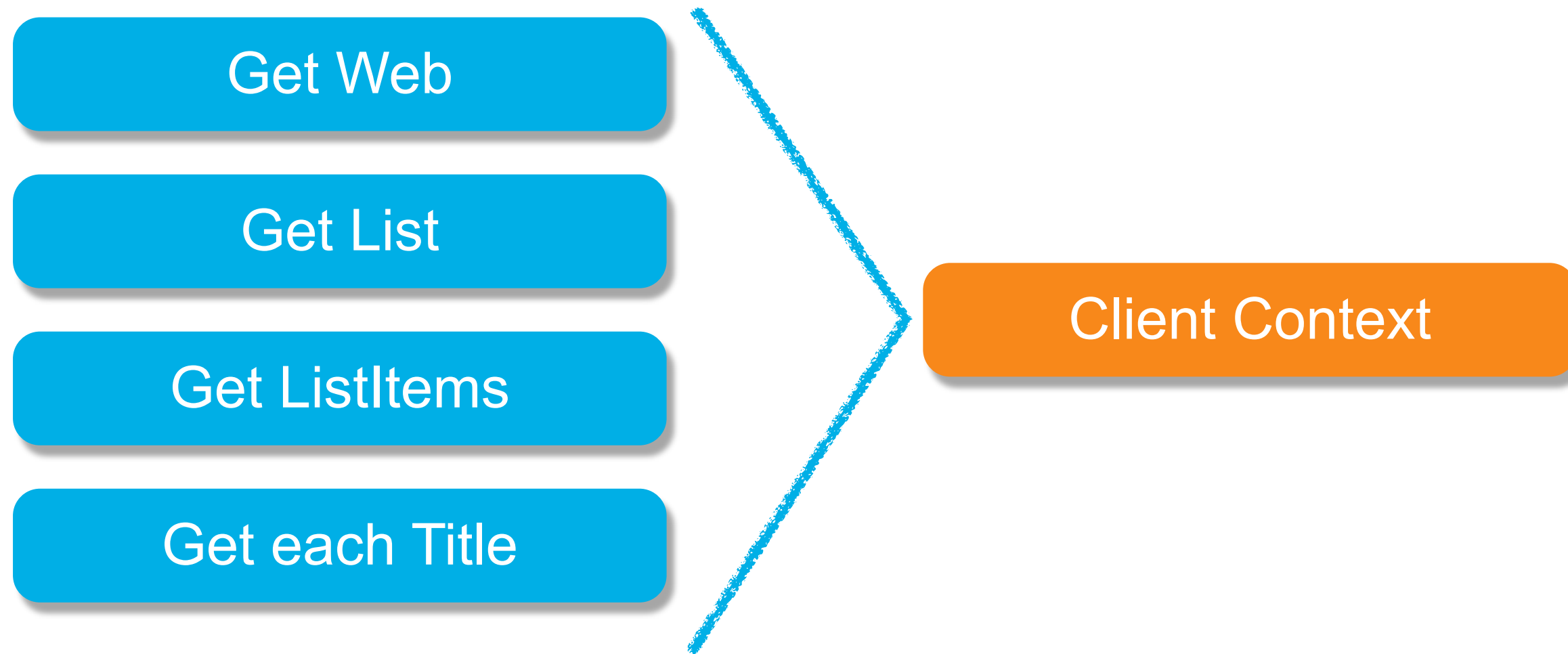
Get Web

Get List

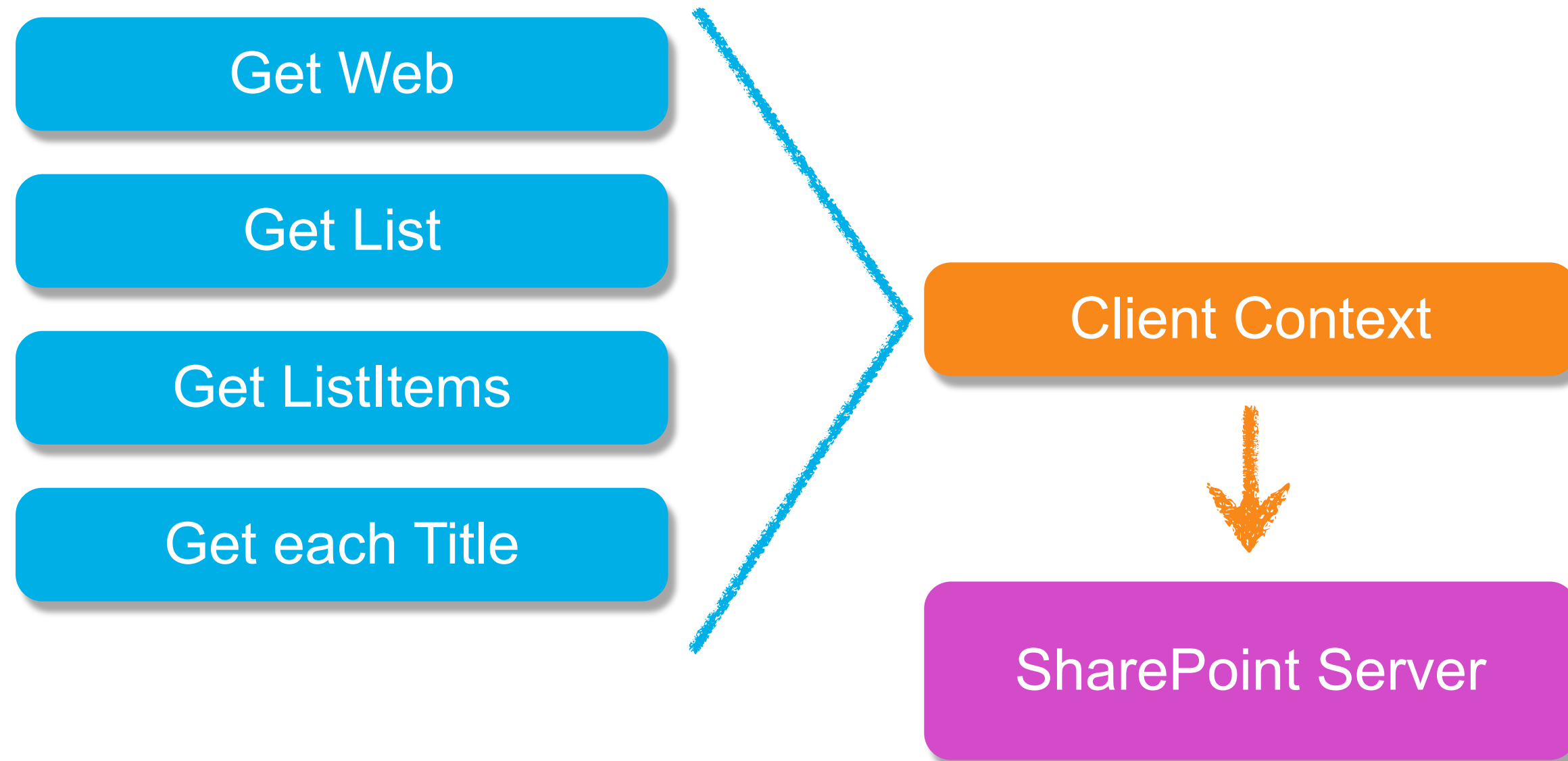
Get ListItems

Get each Title

Recap - Request Batching



Recap - Request Batching




Recap - Request Trimming

- By default all element properties are returned
- limit these properties by extending

`ClientContext.load()` call

- New capabilities in 2013
 - Social API
 - Taxonomy
 - UserProfile API extended
 - Analytics / Site Health Status





Let's do some JSOM

SharePoint REST API

- Supported by
 - SharePoint OnPremise
 - SharePoint Online (Office 365)

REST, again?

Yes, because of ...

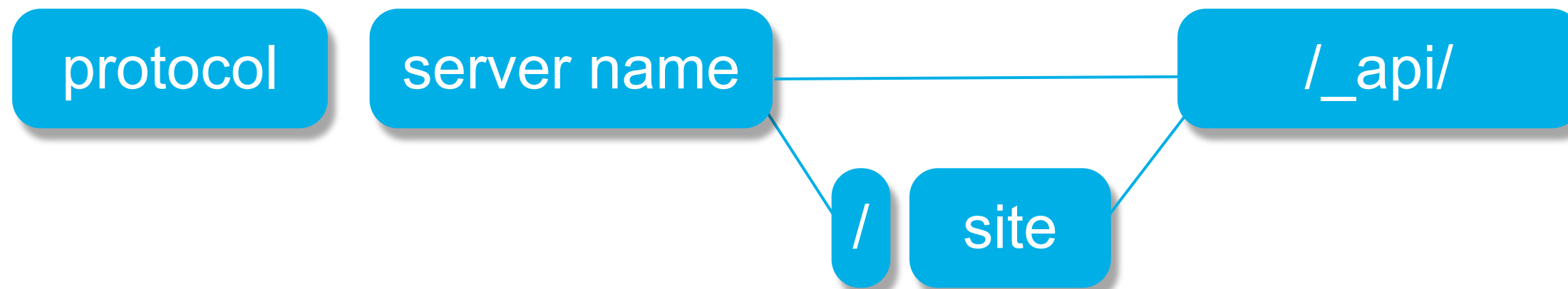
REST, again?

Yes, because of ...

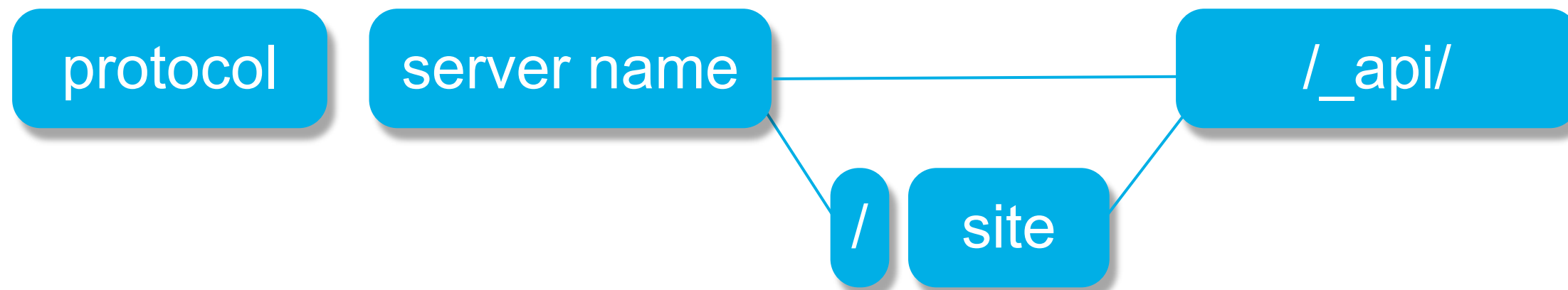
- REACH
 - Gain more SharePoint devs
- STANDARDS
 - SharePoint is offering API's using Web Standards
- FLEXIBILITY
 - Use your favorite development language

REST Uri Syntax

REST Uri Syntax

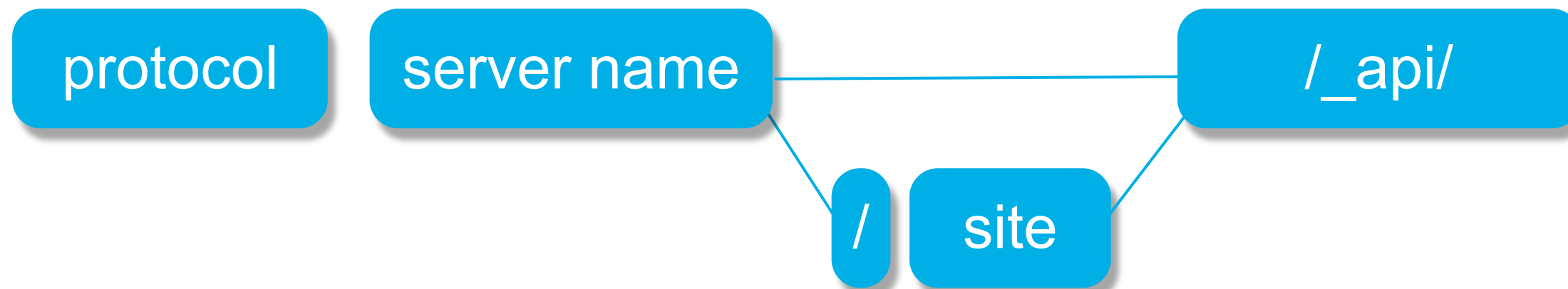


REST Uri Syntax



http://my-sharepoint/_api/

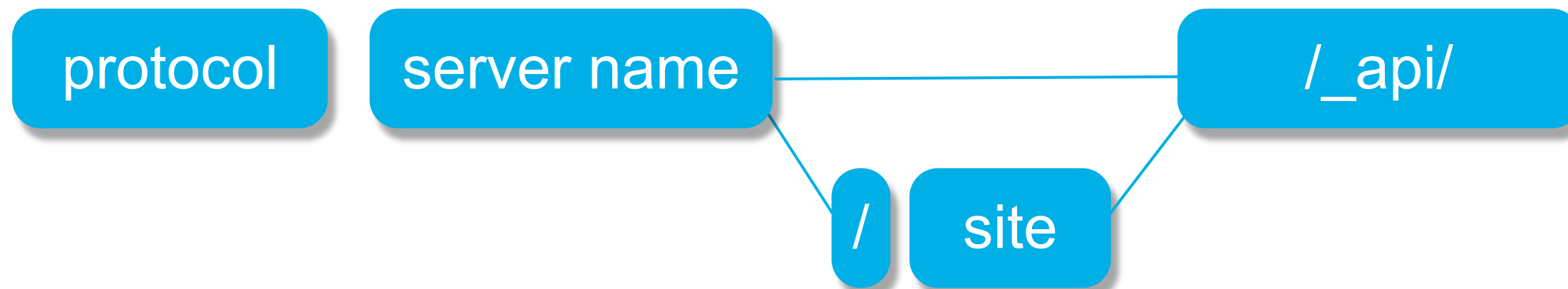
REST Uri Syntax



`http://my-sharepoint/_api/`

`https://dotnetrocks.sharepoint.com/_api/`

REST Uri Syntax



`http://my-sharepoint/_api/`

`https://dotnetrocks.sharepoint.com/_api/`

`http://my-sharepoint/sites/hr/_api/`

REST Namespaces

Maps the SharePoint structure

- /_api/site
- /_api/web
- /_api/web/lists
- /_api/web/lists('guid')
- /_api/web/lists('guid')/items
- /_api/search/
- /_api/publishing/
- /_api/userprofiles/

HTTP Method mapping

Fits perfectly to SharePoint's habits

HTTP Method mapping

Fits perfectly to SharePoint's habits

HTTP POST

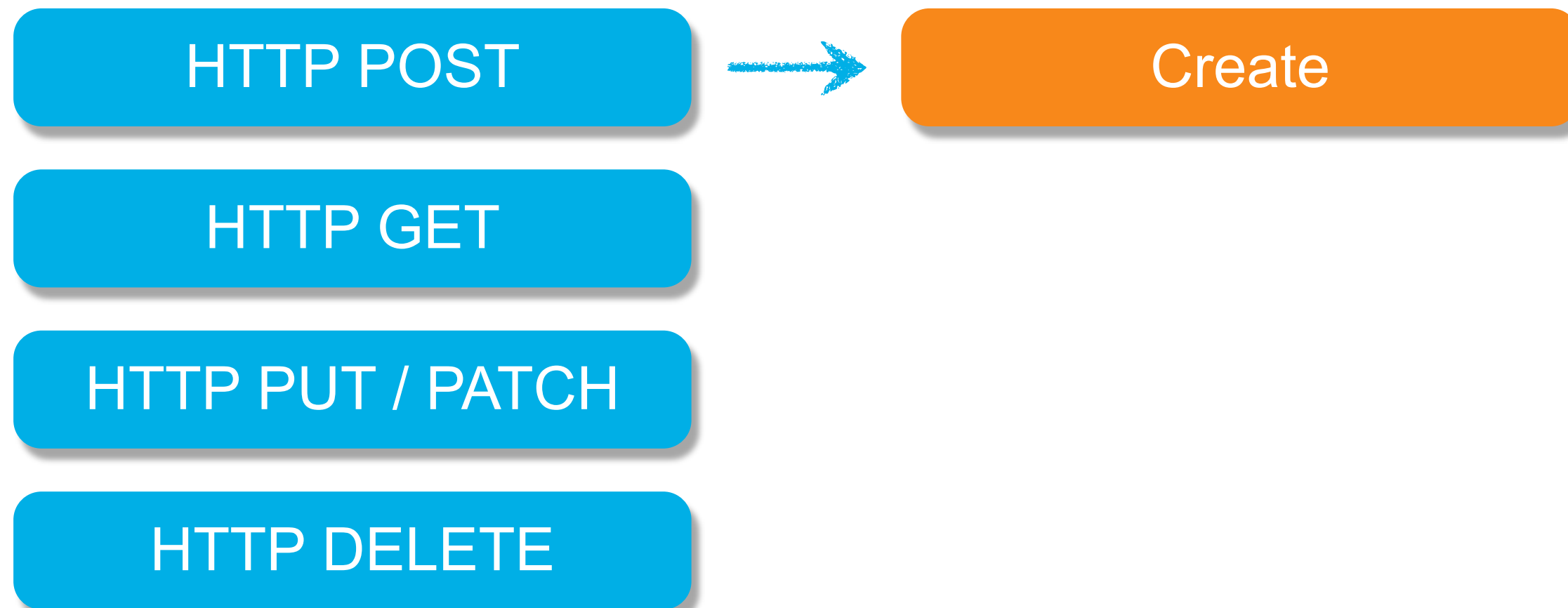
HTTP GET

HTTP PUT / PATCH

HTTP DELETE

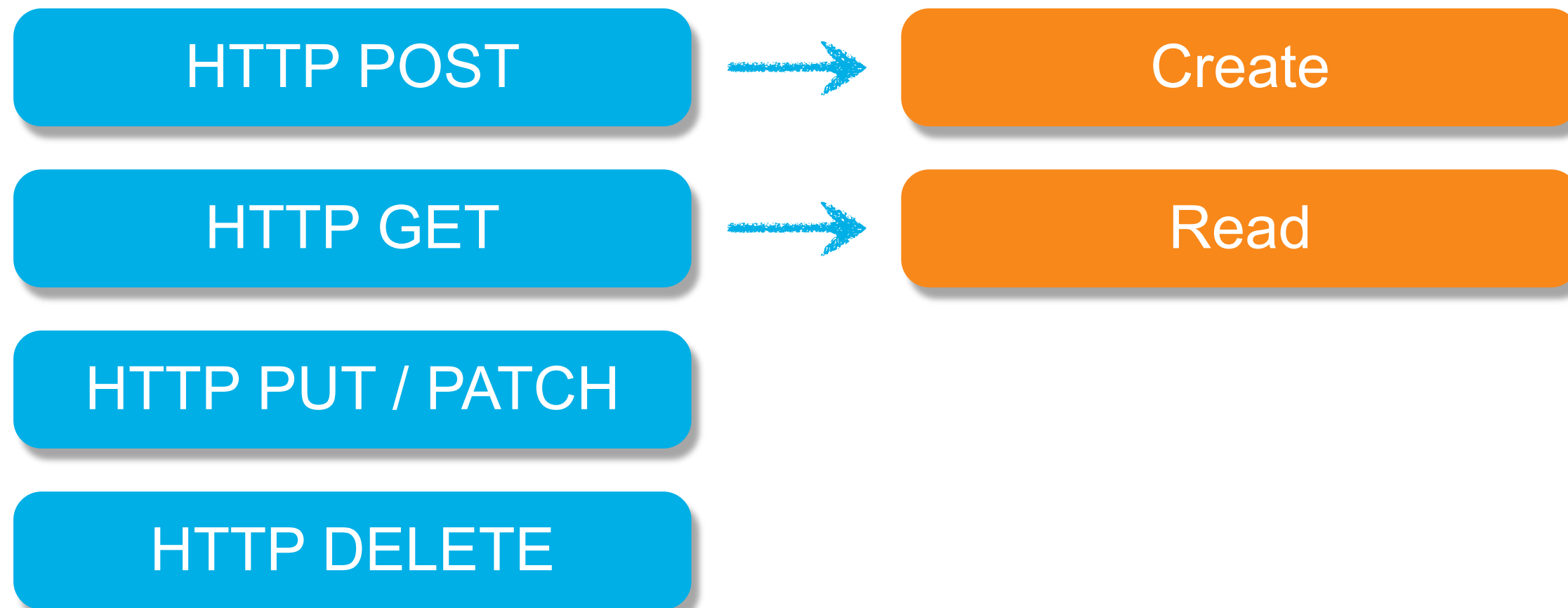
HTTP Method mapping

Fits perfectly to SharePoint's habits



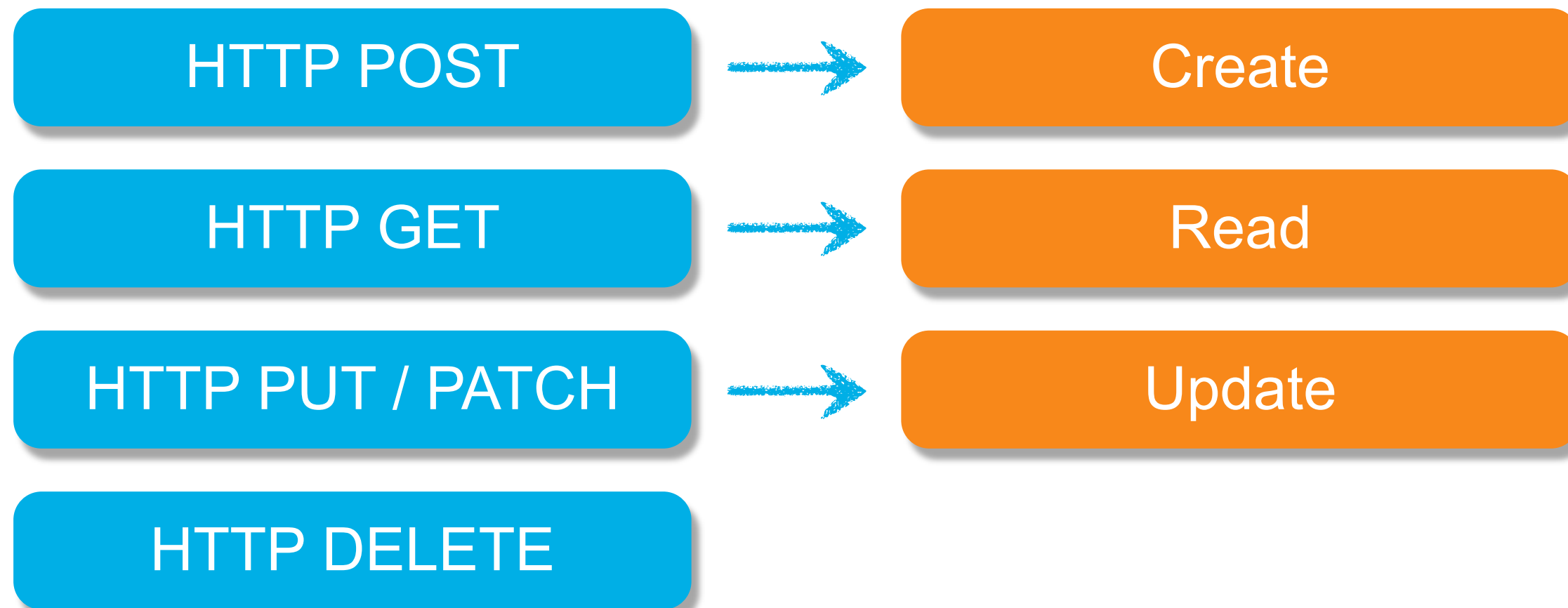
HTTP Method mapping

Fits perfectly to SharePoint's habits



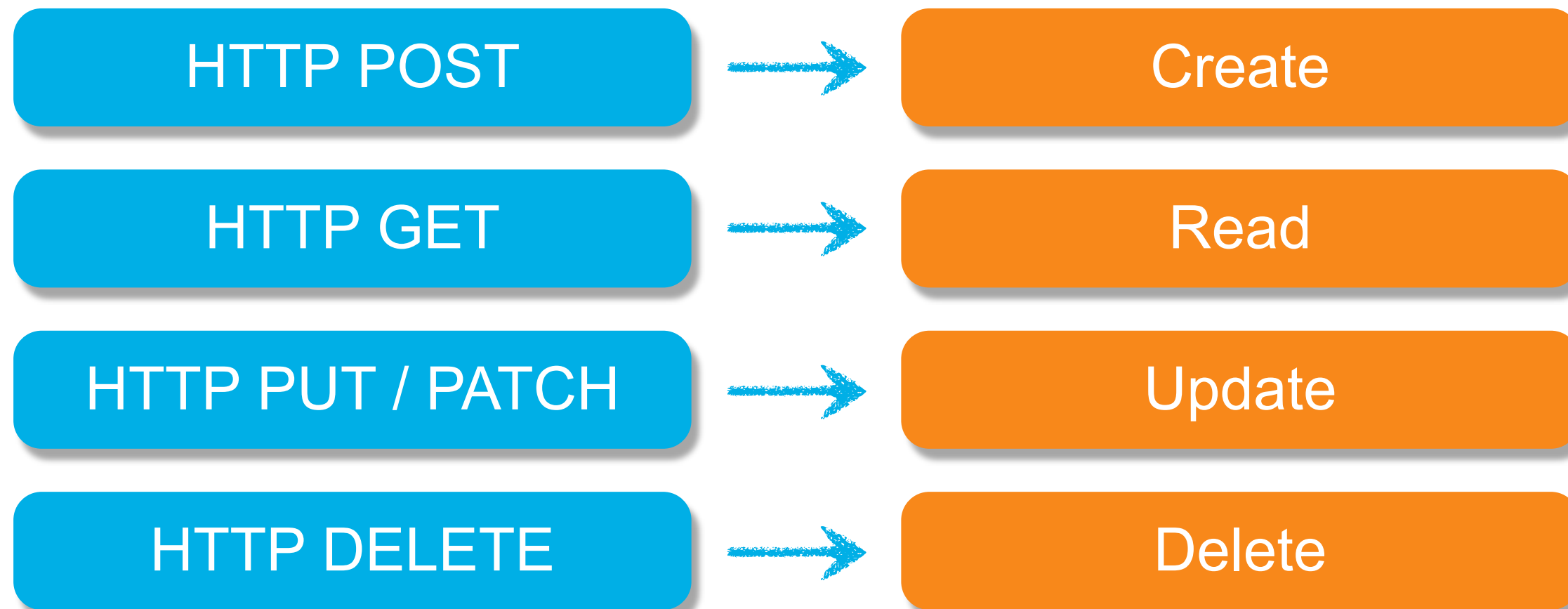
HTTP Method mapping

Fits perfectly to SharePoint's habits



HTTP Method mapping

Fits perfectly to SharePoint's habits





**Let's inspect
some SP-I's**



When to use which API?



Choose your weapon

Working with ListItems / Documents?

Use SharePoint's REST API

Choose your weapon

iOS / Android / LAMP

Use SharePoint's REST API

Choose your weapon

Windows Phone

Mobile Client Object Model

Choose your weapon

HTML5 / JS Apps for SP / Office

Use either JSOM or REST

Choose your weapon

SharePoint TimerJobs / FieldTypes /

WebParts / Admin Stuff?

Server Side OM / PowerShell

Choose your weapon

LOB Silverlight App

Silverlight Object Model

Choose your weapon

...depending on your
development skills!!!

Gimme some
COFFEE



CoffeeScript

Learn **CoffeeScript** (or TypeScript) to make your
client-side code more **robust**

CoffeeScript

CoffeeScript compiles into JavaScript

<http://www.coffeescript.org>



Why CoffeeScript?!

All code samples are online
for free at github

<http://github.com/ThorstenHans/Conferences/>

Thx for your attention! ;-)





EXPERTS INSIDE