

Frontend Middleware

Rico Pfaus | GALERIA Kaufhof

who**ami**:://



- Interface Dev und Frontend Lead im Bereich eShop-Systeme bei GALERIA Kaufhof
- Webdeveloper aus Leidenschaft seit 2000
- loving all things JS/ES/TS/coffee
- done too much affiliate integration ;)

PART#1

a **JUMP** back to 2014

Stichwort **Vertikalisierung**

- Vertikaler Schnitt der Applikation in Microservices auf Basis fachlicher Zuständigkeit
- inhaltliche, organisatorische und teilweise räumliche Trennung der agilen Teams
- „Horizontale“ Frontend in einer vertikalen Welt
- Strikte Trennung von Frontend / Backend Devs

Fachliche Probleme

- koordinierte Einbindung von Third-Parties (z.B. Tracking, Affiliates, ...) in 5 autonomen Vertikalen ist komplex und schwer synchronisierbar
- Fragmentierung und Duplikation von Stories und Know-How
- fehlende fachliche Verantwortung
- kein dediziertes Team für „Tracking & Co.“

Technische Probleme

- „Klassische Integration“ von Third-Parties nicht professionell einsetzbar (Global, Blocking, Inline, ...)
- kein Unittesting von Snippets, somit keine volle QA
- keine klaren Konventionen
- kein langfristiges Konzept

PART#2

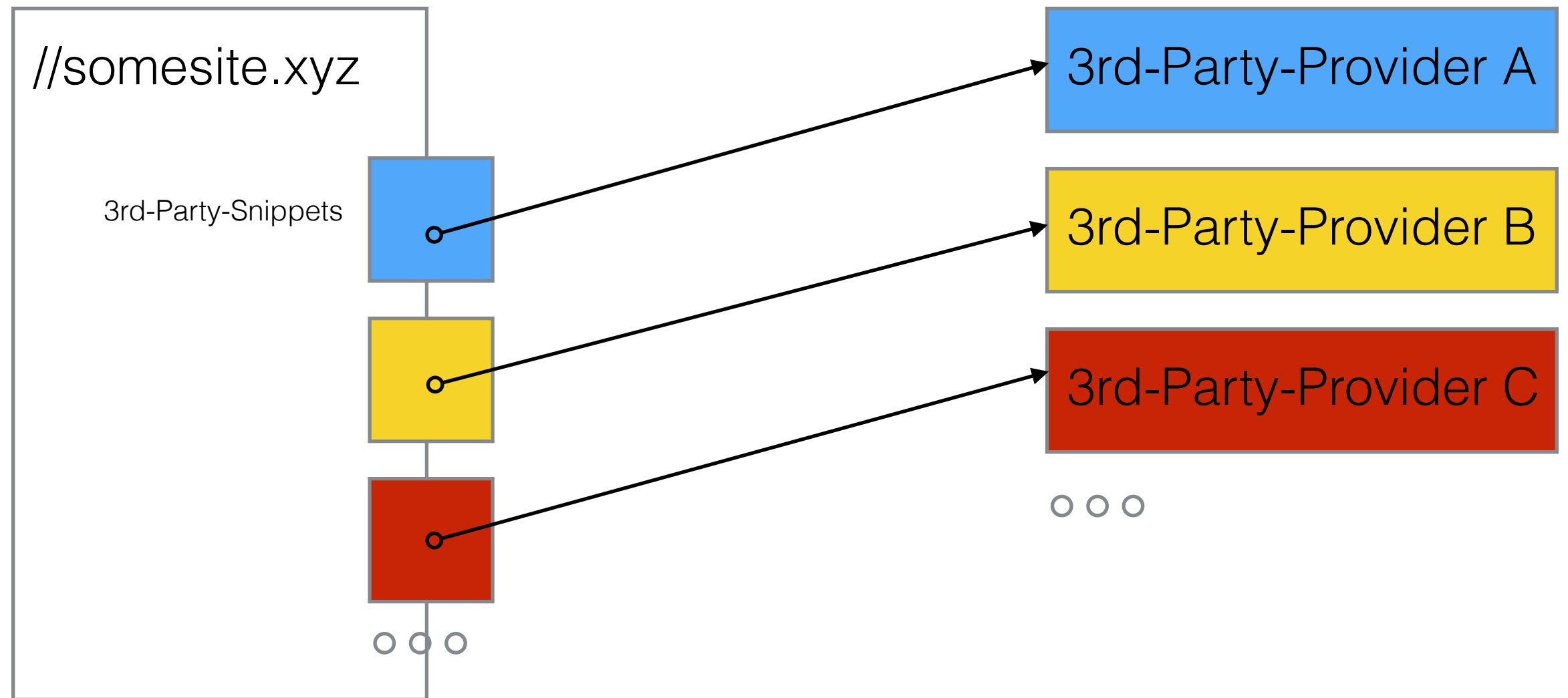
Third-Party-Integration **2015**

Variante#1: Einbindung nach **Vorschrift**

Beispiel „etracker“: synchron, blocking, global

```
1 <script type="text/javascript" charset="UTF-8">
2 var et_pagename = "Tracking";
3 var et_areas = "de/Advertiser/Plattform-1";
4 var et_tag = "language={de-DE}";
5 var et_ilevel = "4";
6 var et_url = "https%3a%2f%2faffili.net%2fde%2fAdvertiser%2fplattform%2ftracking";
7 </script>
8 <script id="_etLoader" type="text/javascript" charset="UTF-8"
9 data-secure-code="0Ymp83" src="//static.etracker.com/code/e.js"></script>
10 <noscript><link rel="stylesheet" media="all" href="//www.etracker.de/cnt_css.php?et=0
11 |
```



Variante#1: Einbindung nach **Vorschrift**




Variante#1: Einbindung nach **Vorschrift**

- häufig blockierende Einbindung: erzeugt enorme Gefahr von schlechter Performance, verursacht durch Drittsysteme
- Inline-Hölle: Javascript steckt in HTML-Templates
- nicht sauber testbar (höchstens holprig funktional)
- globale Variablen
- nicht modular


Variante#2: Einbindung per **Tagmanager**




Settings




Tags




Serve Rules




Templates




Variables




Test Mode




Build Info



Re-Build Tags



Change Log




Reports


Management


Publishing

Performance

Opened Tabs



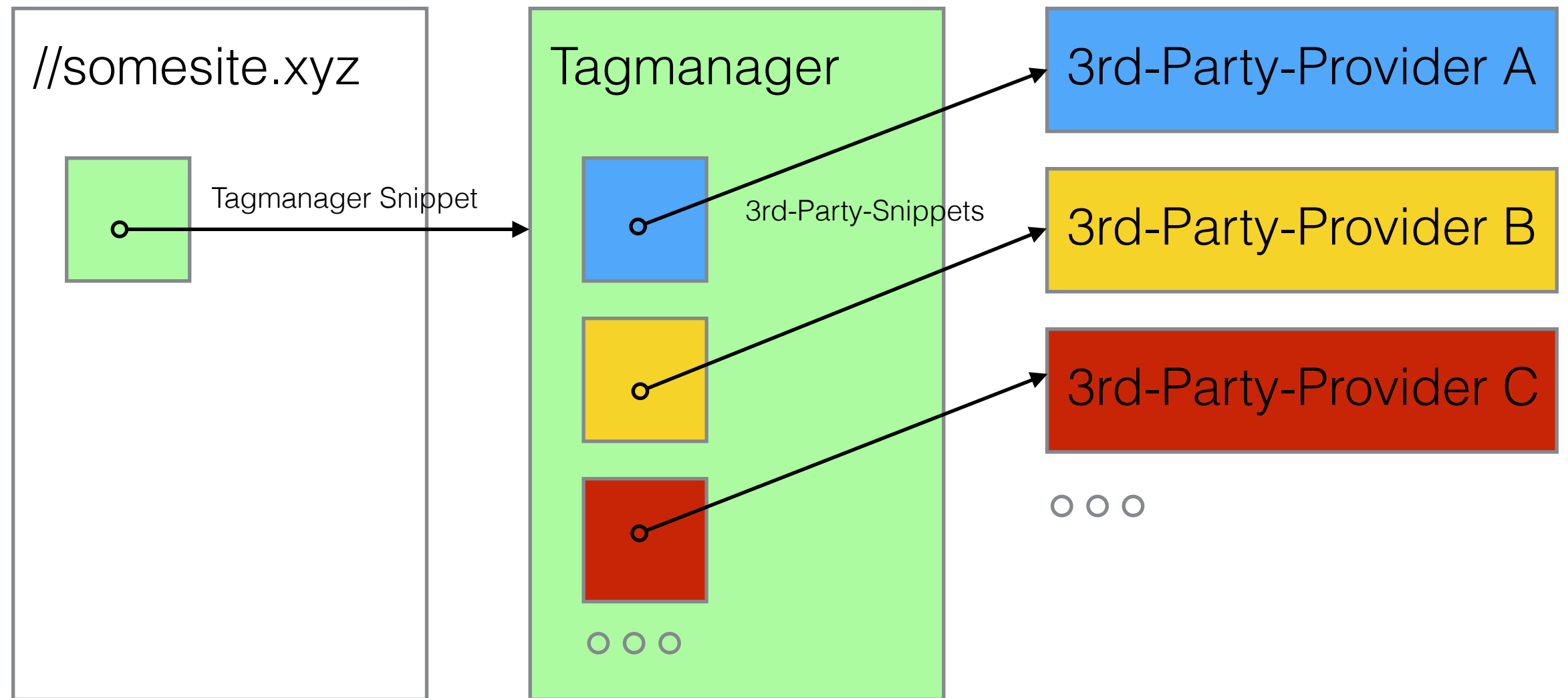




Tags for **www.galeria-kaufhof.de**

Tag Name	Type	Description	Build Priority	Last Modified
Basket: Performance Technologies	Non-conversion		None	07/Oct/2014
Basket: Sociomantic	Non-conversion	(No Description)	None	22/Jul/2013
Basket: Tradedoubler	Non-conversion		None	06/Aug/2013
Basket: Xplosion	Non-conversion	(No Description)	None	17/Jul/2013
Basket: Zanox Master Tag	Non-conversion	(No Description)	None	09/Dec/2013
Category: Affilinet	Non-conversion	Affilinet category tracking.	None	05/Aug/2013
Category: Criteo	Non-conversion	(No Description)	None	29/Jan/2014
Category: NextPerformance	Non-conversion		500	11/May/2015
Category: Performance Technolo...	Non-conversion		None	07/Oct/2014
Category: Sociomantic	Non-conversion	(No Description)	None	29/Jul/2013
Category: Tradedoubler	Non-conversion	(No Description)	None	06/Aug/2013
Category: Xplosion	Non-conversion	(No Description)	None	29/Jul/2013
Category: Zanox MasterTag	Non-conversion	(No Description)	None	09/Dec/2013

Variante#2: Einbindung per **Tagmanager**



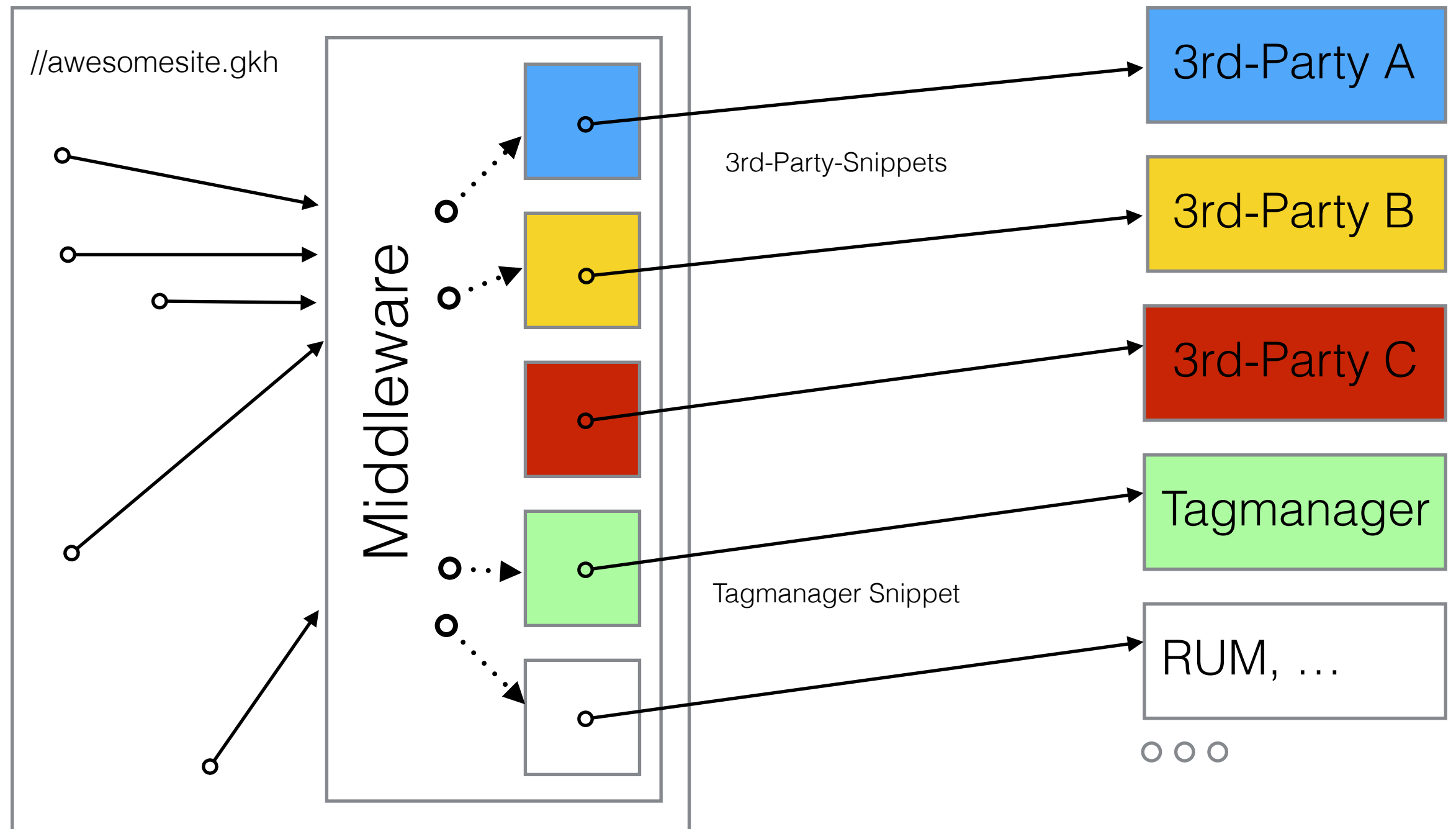
Variante#2: Einbindung per **Tagmanager**

- Tagmanager sind selber Third-Party-Software
- sehr einfach blockierbar durch Adblocker, was zu vollständigem Verlust aller Trackinginfos führt
- Gefahr des Wildwuchses durch Pflege in Fachbereichen
- Snippets nicht testbar
- Code nicht in Repository bzw. keine Integration in Build-Pipeline

PART#3

The way it **should** be ...

Variante#3: Die **Perfekte** Welt



Variante**#3**: Die **Perfekte** Welt

- Frontendlogik nimmt (an jeder Stelle im Markup) Informationen entgegen und spielt diese regelbasiert an interessierte Abnehmer aus
- strikte Konventionen zu erforderlichen Daten und Informationen je Seitentyp
- Third-Party-Snippets liegen im Repository und werden in CI/CD-Pipeline automatisiert Unit-getestet und deployed
- alle Third-Party-Skripte on-the-fly austauschbar

PART#4

Middleware insights

Middleware insights: Konzept

- Modulare Architektur auf Basis von AMD
- Third-Party-Code ist vollständig in Services (bzw. Plugins) gekapselt
- regelbasierte Ausspielung von Services auf Basis der übergebenen Daten
- Datenübergabe zur Renderzeit per meta-Tags
- Datenübergabe zur Laufzeit per API (One-Way Broadcast an die Services)

Middleware insights: Datenübergabe

Übergabe von Daten zur Renderzeit per meta-Tag

```
<meta name="gk:dal:data" content='{
  .."site":{
    ...."id":"jump_live"
  },
  .."page":{
    ...."type":"category",
    ...."name":"Experimentieren & Entdecken"
  },
  .."category":{
    ...."id":"/spielzeug/spielwelten/experimentieren-entdecken/",
    ...."name":"Experimentieren & Entdecken",
    ...."productIds":[123, 456, 789]
  },
  .."user":{
    ...."id":"",
    ...."session":""
  }
}' />
```

Übergabe der Kategorieinfos
an die Middleware

Middleware insights: Datenübergabe

Verarbeitung der Daten innerhalb der Services

```
define "ba/lib/dal/aff/xplosion", ["gk/globals/window", "gk/vendor/cookie", "gk/lib/logger"], (window, cookie, Logger) ->
  ..logger = new Logger("ba/lib/dal/aff/xplosion")

  ..class Xplosion

  ....constructor: (@dal, @data, @config) ->
  .....# build params object
  .....customerID = "galeria-kaufhof.de"
  .....params = (switch @data.page.type
  .....when "homepage"
  .....{
  .....  event_id: "homepage"
  .....}
  .....when "category"
  .....{
  .....  event_id: "shop_visit"
  .....  shop_id: @data.category.id
  .....  shop_trackingproducts: @data.category.productIds
  .....}
  .....when "search"
  .....{
  .....  event_id: "search_result"
  .....  user_searchkeywords: @data.search.keywords
  .....  shop_trackingproducts: @data.category.productIds
  .....}
```

Weiterreichen der Kategorieinfos
an 3rd-Party-Snippet

Middleware insights: Datenübergabe

Übergabe von Daten zur Laufzeit per API

"use strict"

```
define "or/controller/cartEntryLayer", ["jquery", "gk/lib/DAL", "gk/lib/metaData"], ($, DAL, md) ->
  class CartEntryLayer
    constructor: (@$el) ->
      $(document).trigger("or:minicart:changed")

      dalData = md.collectJson("gk:dal:data", @$el)

      DAL.broadcast "addtocart", {
        layer: @$el,
        product: dalData.product
      } if dalData?.product?

      @$el.on "click", "#or-cartentry__back", (e) =>
        e.preventDefault()
        @$el.trigger("gk:layer:close")

    destroy: ->
      @$el.off()
```

Übergabe der Produktdaten beim AddToCart zur Laufzeit per API

Middleware insights: Services

regelbasierte Einbindung der Services

```
# DAL rules configuration
DAL_RULES =
... "ba/lib/dal/bsna":
...   test: true
...   rule: (->true)
... "ba/lib/dal/econda": true
... "ba/lib/dal/richrelevance": true
... "ba/lib/dal/stacy": true
... "ba/lib/dal/tagmanager": true
... # facebook pixels
... "ba/lib/dal/aff/facebookWCA": (data) ->
...   data.page.type in DAL_TRACKING_PAGETYPES
... "ba/lib/dal/aff/facebookConversion": (data) ->
...   data.page.type is "checkout-confirmation"
... # A/B test for testing criteo vs. xplosion
... "ba/lib/dal/aff/criteo": (data) ->
...   data.page.type in DAL_TRACKING_PAGETYPES and abTest(50, 365) is "A"
... "ba/lib/dal/aff/xplosion": (data) ->
...   data.page.type in DAL_TRACKING_PAGETYPES and abTest(50, 365) is "B"
```

Middleware insights: Services

Beispiel facebookConversion-Pixel

```
11 define "ba/lib/dal/aff/facebookConversion", ["gk/globals/window", "gk/lib/logger"], (window, Logger) ->
12 ..logger = new Logger("ba/lib/dal/aff/facebookConversion")
13
14 ..class FacebookConversion
15
16 ....constructor: (@dal, @data, @config) ->
17 .....# load FB pixel and append it to DOM
18 .....if not window._fbq
19 .....  window._fbq = []
20 .....  _fbq = window._fbq
21 .....  if not _fbq.loaded
22 .....    fbds = window.document.createElement('script')
23 .....    fbds.src = "//connect.facebook.net/en_US/fbds.js"
24 .....    s = window.document.getElementsByTagName("script")[0]
25 .....    s.parentNode.insertBefore(fbds, s)
26 .....    _fbq.loaded = true
27 .....    _fbq.push(["addPixelId", "6020827373384"])
28 .....    _fbq.push(["track", "PixelInitialized", {}])
29 .....# track event depending on type
30 .....if @data.page.type is "checkout-confirmation"
31 .....  ids = []
32 .....  for p,i in @data.order.products
33 .....    ids.push(p.ean)
34 .....    _fbq.push(["track", "Purchase", {
35 .....      value: @data.order.priceData.total
36 .....      currency: "EUR"
37 .....      content_ids: ids
38 .....      content_type: "product"
39 .....    }])
```


Middleware insights: Services

facebookConversion-Pixel Unittest

```
65 ..it "should pass the correct account to FB", ->
66   ...new service(dalApi, gkDalDataMock)
67   ...assert.isTrue(hasSubArray(window._fbq, ["addPixelId", "6020827373384"]))
68
69 ..it "should set the FB pixel as initialized", ->
70   ...new service(dalApi, gkDalDataMock)
71   ...assert.isTrue(hasSubArray(window._fbq, ["track", "PixelInitialized", {}]))
72
73 ..it "should not send any product data when the pagetype isnt 'checkout-complete'", ->
74   ...gkDalDataMock.page.type = "homepage"
75   ...new service(dalApi, gkDalDataMock)
76   ...assert.equal(window._fbq.length, 2) # -> PixelId, Initialized
77
78 ..it "should pass all product ids and the total price when the pagetype is 'checkout-confirmation'", ->
79   ...gkDalDataMock.page.type = "checkout-confirmation"
80   ...gkDalDataMock.order =
81     ...products: [
82       ...{ean: 4000000000001}
83       ...{ean: 4000000000002}
84       ...{ean: 4000000000003}
85     ...]
86     ...priceData:
87       ...total: 123.45
88   ...new service(dalApi, gkDalDataMock)
```


Middleware insights: Konventionen

- Klare, übergreifende Konventionen für Datentypen
- Virtuelle Typisierung per JSON-Schema
- Möglichkeit zur automatisierten Validierung der applikationsseitig übergebenen Daten (z.B. per Selenium)

Middleware insights: Konventionen

Typkonvention am Beispiel *DALCategoryData*

```
DALCategoryData = {  
  /**  
   * ID der Kategorie (z.B. URL-Pfad o.ä.)  
   */  
  "id" : {  
    "type" : "String",  
    "mandatory" : true  
  },  
  /**  
   * Anzeigename der Kategorie (wie z.B. textuell in der Navigation dargestellt)  
   */  
  "name" : {  
    "type" : "String",  
    "mandatory" : true  
  },  
  /**  
   * Array mit Produkten in dieser Produktliste, beschnitten auf maximal die Top-36).  
   */  
  "productIds" : {  
    "type": "Array:int",  
    "mandatory" : true,  
    "apiVersion": 2  
  }  
}
```

Da war ja **noch was**

... das Projekt wird natürlich **Open-Source!**



(Hatte ich auf der FrOSCon leider völlig vergessen zu erwähnen)

That's it **so far**

Any **questions?**

Thanks for your attention!

 <http://github.com/ryx>

 [@ryxxxx](#)