



adaptTo()

EUROPE'S LEADING AEM DEVELOPER CONFERENCE
28th – 30th SEPTEMBER 2026

Tracking Through the Transition

Janic Wyslich, Martin Schröter

BigCompany after website relaunch for global brand website

IT – DEPARTMENT



It was sometimes hard
but finally a success...

MARKETING



How can we measure
campaign success?
Does traffic increase
now?

...

PREQUEL





Janic Wyslich, diva-e



Martin Schröter, diva-e

Why is that ?

- Budget cuts, scope-outs
- The promise of Tagmanagers
- Assumption of OOTB

“Can we do the MarTech integration before go-live
... next week?
...its simple, just inject the Tagmanager “

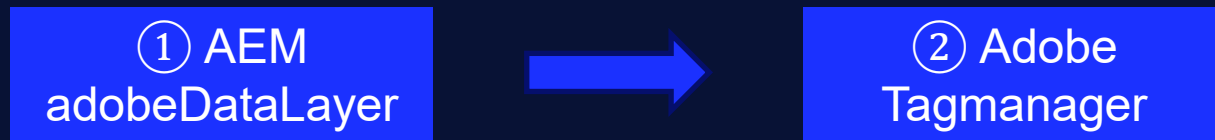
PREQUEL

It can be simple, right!

But mostly it's not that simple!

But it is still simple if you
follow best practices

The Tagmanager can only work on the data it's fed



MARTECH 101 – ① AEM Integration

1. Use core components dataLayer

Data Layer

Configure support for Adobe Client Data Layer

☐ ENABLE PROPERTY INHERITANCE ⓘ

PROPERTY	VALUE	DESCRIPTION	INFO
Data Layer client library not included	☐		ⓘ
Data Layer enabled	☒		ⓘ
Data Layer object name	adobeDataLayer		

MARTECH 101 – ① AEM Integration

1. Use core components dataLayer
2. Use adobeDataLayer

Adobe Data Collection

Configure script for Adobe Data Collection integration

☐ ENABLE PROPERTY INHERITANCE ⓘ

PROPERTY	VALUE	DESCRIPTION	INHERITED	OVERRIDDEN
Activate Data Collection	☒	ⓘ	☐	☐
Activate Pre-Hiding Snippet for Target	☒	ⓘ	☐	☐
Activate on author instance (for debugging purposes only)	☒	ⓘ	☐	☐
Load without consent	☐	ⓘ	☐	☐
Consent category	C0002	ⓘ	☐	☐
Adobe Data Collection script URL	//assets.adobedtm.com/e3425ae...	ⓘ	☐	☐
FPID: extra fetch per browser session	☐	ⓘ	☐	☐

MARTECH 101 – ① AEM Integration

1. Use core components dataLayer
2. Use adobeDataLayer
3. Use proper consent integration

OneTrust configuration

Configure OneTrust configuration

☐ ENABLE PROPERTY INHERITANCE ⓘ

PROPERTY	VALUE	DESCRIPTION	INHERITED	OVERRIDDEN
Activate OneTrust	☒	ⓘ	☐	☐
Activate OneTrust Autoblock	☐	ⓘ	☐	☐
OneTrust Domain ID	3f7ee518-e94e-48d1-...	ⓘ	☐	☐
SRI for OneTrust SDK script		ⓘ	☐	☐
SRI for OneTrust Autoblock script		ⓘ	☐	☐
Third Party Notification Experience Fragment	/content/experience-fragments ... ⓘ	ⓘ	☐	☐
OneTrust DataLayer Name	adobeDataLayer	ⓘ	☐	☐
OneTrust Language Override		ⓘ	☐	☐
[DEBUG] - Activate OneTrust on author instance	☐	ⓘ	☐	☐

MARTECH 101 – ① AEM Integration

1. Use
core components
dataLayer
2. Use
adobeDataLayer
3. Use
proper consent
integration
4. Use
data-cmp-datalayer

```
<button class="accordion-expand-all"  
  data-sly-use.dataLayerButton="${'eu.man.aem.datalayer.GenericComponentDataLayer' @  
    injectTitle='Expand/Collapse all',  
    injectId='btn',  
    prefixWithParent=true}"  
  data-cmp-data-layer="${dataLayerButton.data.json}"  
  data-cmp-clickable>  
  <span class="cmp-accordion__icon"></span>  
  ${properties.expandCollapseText || 'Expand/Collapse all' @ i18n}  
</button>
```

MARTECH 101 – ① AEM Integration

1. Use core components dataLayer
2. Use adobeDataLayer
3. Use proper consent integration
4. Use data-cmp-datalayer

```
/**
 * Model for the XF contact component
 */
@Model(adaptables = { Resource.class })
public class Contact extends AbstractComponent {

    @ValueMapValue(injectionStrategy = InjectionStrategy.OPTIONAL)
    private String firstName;

    @ValueMapValue(injectionStrategy = InjectionStrategy.OPTIONAL)
    private String lastName;

    @ValueMapValue(injectionStrategy = InjectionStrategy.OPTIONAL)
    private String company;

    /**
     * Retrieves the Component Data
     * @return Component Data
     */
    @Override
    public @NotNull ComponentData getComponentData() {
        return DataLayerBuilder.extending(super.getComponentData()).asComponent()
            .withTitle(() -> StringUtils.joinWith(delimiter: " ", firstName, lastName, company))
            .build();
    }
}
```

MARTECH 101 – ① AEM Integration

1. Use core components dataLayer
2. Use adobeDataLayer
3. Use proper consent integration
4. Use data-cmp-datalayer
5. Please don't break core/wcm/components/page/v3/page

```

1  <!--/*
2  ~ Copyright 2021 Adobe
3  ~
4  ~ Licensed under the Apache License, Version 2.0 (the "License");
5  ~ you may not use this file except in compliance with the License.
6  ~ You may obtain a copy of the License at
7  ~
8  ~     http://www.apache.org/licenses/LICENSE-2.0
9  ~
10 ~ Unless required by applicable law or agreed to in writing, software
11 ~ distributed under the License is distributed on an "AS IS" BASIS,
12 ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 ~ See the License for the specific language governing permissions and
14 ~ limitations under the License.
15 ~~~~~*/-->
16 <!DOCTYPE HTML>
17 <html data-sly-use.page="com.adobe.cq.wcm.core.components.models.Page" lang="{page.language}"
18     data-sly-use.pwa="com.adobe.cq.wcm.core.components.models.PWA"
19     data-sly-use.head="head.html"
20     data-sly-use.footer="footer.html"
21     data-sly-use.redirect="redirect.html">
22 <head data-sly-call="{head.head @ page = page, pwa = pwa}"></head>
23 <body class="{page.cssClassNames}"
24     id="{page.id}"
25     data-cmp-link-accessibility-enabled
26     data-cmp-link-accessibility-text="{ 'opens in a new tab' @ i18n}"
27     data-cmp-data-layer-enabled="{page.data ? true : false}"
28     data-cmp-data-layer-name="{page.dataLayerName @ context='scriptToken'}">
29 <script data-sly-test.dataLayerEnabled="{page.data}">
30     var dataLayerName = '{page.dataLayerName @ context="scriptToken"}' || 'adobeDataLayer';
31     window[dataLayerName] = window[dataLayerName] || [];
32     window[dataLayerName].push({
33         page: JSON.parse("{page.data.json @ context='scriptString'}"),
34         event: 'cmp:show',
35         eventInfo: {
36             path: 'page.{page.id @ context="scriptString"}'
37         }
38     });
39 </script>
40 <sly data-sly-test.isRedirectPage="{page.redirectTarget && (wcmmode.edit || wcmmode.preview)}">

```

MARTECH 101 – ① AEM Integration

1. Use core components dataLayer
2. Use adobeDataLayer
3. Use proper consent integration
4. Use data-cmp-datalayer
5. Please don't break core/wcm/components/page/v3/page
6. Setup frontend

```
export type DataLayerEvent = {
  eventName?: string;
  type: 'component' | 'page' | 'user' | undefined;
  id: string;
  data: Record<string, unknown>;
};

Martin Schröter, 2 months ago | author (Martin Schröter)
export class DataLayerService {
  private pageID;
  private alloyMonitoringRegistered = false;

  constructor() {
    this.pageID = this.set_page_id();
    this.handleConsentChanges();
  }

  > async getState(): Promise<unknown> { ...
  }

  push(dataLayerEvent: DataLayerEvent): void {
    const event = dataLayerEvent.eventName;
    const eventInfo = this.buildEventInfo(dataLayerEvent.type, dataLayerEvent.id);
    const update = this.nester(eventInfo?.eventInfo?.reference, this.prepareTrackingEvent(dataLayerEvent.data));

    const pushObject: Record<string, unknown> = {
      ...(event && { event }),
      ...eventInfo,
      ...update,
    };

    window.adobeDataLayer = window.adobeDataLayer || [];
    window.adobeDataLayer.push(pushObject);
  }
}
```

MARTECH 101 – ② Tagmanager Setup

The problem:

Tagmanagers are in fact
Spaghetti-Code
Generators

They grow fast until:

- Unmaintainable
- Untestable
- Unpredictable

Edit Code (JavaScript)

```
1 const currencyCode = _satellite.getVar('Page: Currency Code');
2 const fullStateEvent = adobeDataLayer.getState();
3 const pageType = _satellite.getVar('Page: Type');
4 const page = adobeDataLayer.getState().page[Object.keys(adobeDataLayer.getState().page)[0]];
5 const eventMapping = {
6   Productlistingpage: 'scOpen,event23,event35,event36',
7   Servicepage: 'prodView,event24,event35,event36',
8   Productpage: 'prodView,event24,event35,event36',
9   Shopcheckoutpage: 'scView,event35,event36'
10 };
11 const checkoutStep = pageType === 'Shopcheckoutpage' ? event.message?.component?.cart?.checkoutStep : '';
12 const ecommerceStepValue = _satellite.getVar('Shop: E-Commerce Step')(pageType, checkoutStep);
13 const pageProducts = contilib.getPageViewProducts(pageType, page, fullStateEvent);
14
15 s.events = s.apl(s.events, eventMapping[pageType] || '');
16
17 if (pageProducts.length > 0) {
18   if (currencyCode === 'CHF') {
19     s.linkTrackEvents = s.apl(s.linkTrackEvents, 'event101');
20     s.events = s.apl(s.events, 'event101');
21   }
22
23   if (currencyCode === 'NOK') {
24     s.linkTrackEvents = s.apl(s.linkTrackEvents, 'event103');
25     s.events = s.apl(s.events, 'event103');
26   }
```

MARTECH 101 – ② Tagmanager Setup

Solution:

Code Defined Tracking

- acdl_helper library
- XDMtracker plugin
- Transforms a definition into alloy sendEvent requests
- Can be kept in repo
- Testable

```
acdl_helper.xdmtracker.define({
  defaults: {
    eVars: [
      { eVar1: () => document.location.href },
      { eVar2: () => acdl_helper.page.get("dc:title") },
    ],
  },
  events: {
    "my-app:page-load": {
      pageView: true,
      events: ["event1"],
      eVars: [
        { eVar11: "my-app:page-load" },
        { eVar20: (cmp) => cmp["dc:title"] },
      ],
    },
    "my-app:search:error": {
      // pageView omitted → link click
      events: ["event2"],
      eVars: [
        { eVar11: "my-app:search:error" },
        { eVar20: (cmp) => cmp.searchTerm },
      ],
    },
    "my-app:order:complete": {
      eventType: "commerce.purchases",
      events: ["event3"],
      eVars: [
        { eVar11: "my-app:order:complete" },
        { eVar20: (cmp) => cmp.orderId },
      ],
    },
  },
})
```

// optional – base variables for every sendEvent

// required – map of ACDL event names to tracking descriptors

// → eventType defaults to "web.webpagedetails.pageViews"

// → eventType defaults to "web.webInteraction.linkClicks"

// → explicit eventType, no pageView → link click semantics

MARTECH 101 – AI Ready!

One more Benefit!

- As all MARTECH 101 steps are well defined, these are defined as SKILL's.
- As all is in code, as well as the XDM definition, the agent knows exactly what to do.
- This single prompt, creates a feature-branch, does the Sling Model extension, the frontend wiring and the xdm tracker definition update.

We have a new Tracking task: JIRA-1234 wants the following events tracked in the recently merged "ConversionForm" component:

```
const definition = {  
  'form:opened': {  
    webInteractionName: 'formOpened',  
    events: ['event3'],  
    eVars: {  
      eVar11: (cmp) => cmp.context,  
      eVar12: (cmp) => cmp.name,  
      eVar13: (cmp) => cmp.conversion_id  
    },  
  },  
  'form:first-interaction': {  
    webInteractionName: 'firstInteraction',  
    events: ['event2'],  
    eVars: {  

```



Opus 5.5 Mittel



THE MIGRATION PROJECT

MIGRATION CHALLENGE

CMS Migration: **FirstSpirit** to **AEMaaCS**

Analytics Migration: **Piwik PRO** to **Adobe Analytics**

AT THE SAME TIME

RISK

PROJECT DELAY
TRACKING GAPS
LOST COMPERABILITY
MISSING TRUST IN DATA



ONE DATA CONCEPT FOR **PRE** AND **POST** MIGRATION

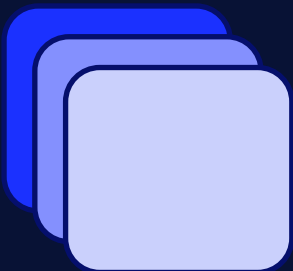
(commit to event and dimension definition from day one)

USE OUR **STANDARD** ANALYTICS IMPLEMENTATION

(data-cmp-datalayer → adobeDataLayer → acdl_helper → xdmtracker)

Solution Design: Anticipate the future

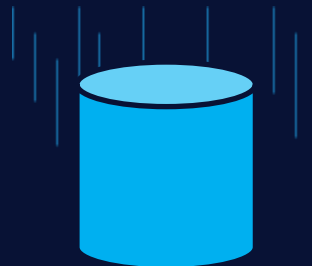
AEM Websites
Countries
Applications
Forms



One
adobeDataLayer

```
{  
  "adobeDataLayer": {  
    "country": "DE",  
    "application": "AEM",  
    "pageType": "homepage",  
    "event": "pageView",  
    "timestamp":  
      "2024-10-01T10:24:56Z"  
  }  
}
```

One
Data Set



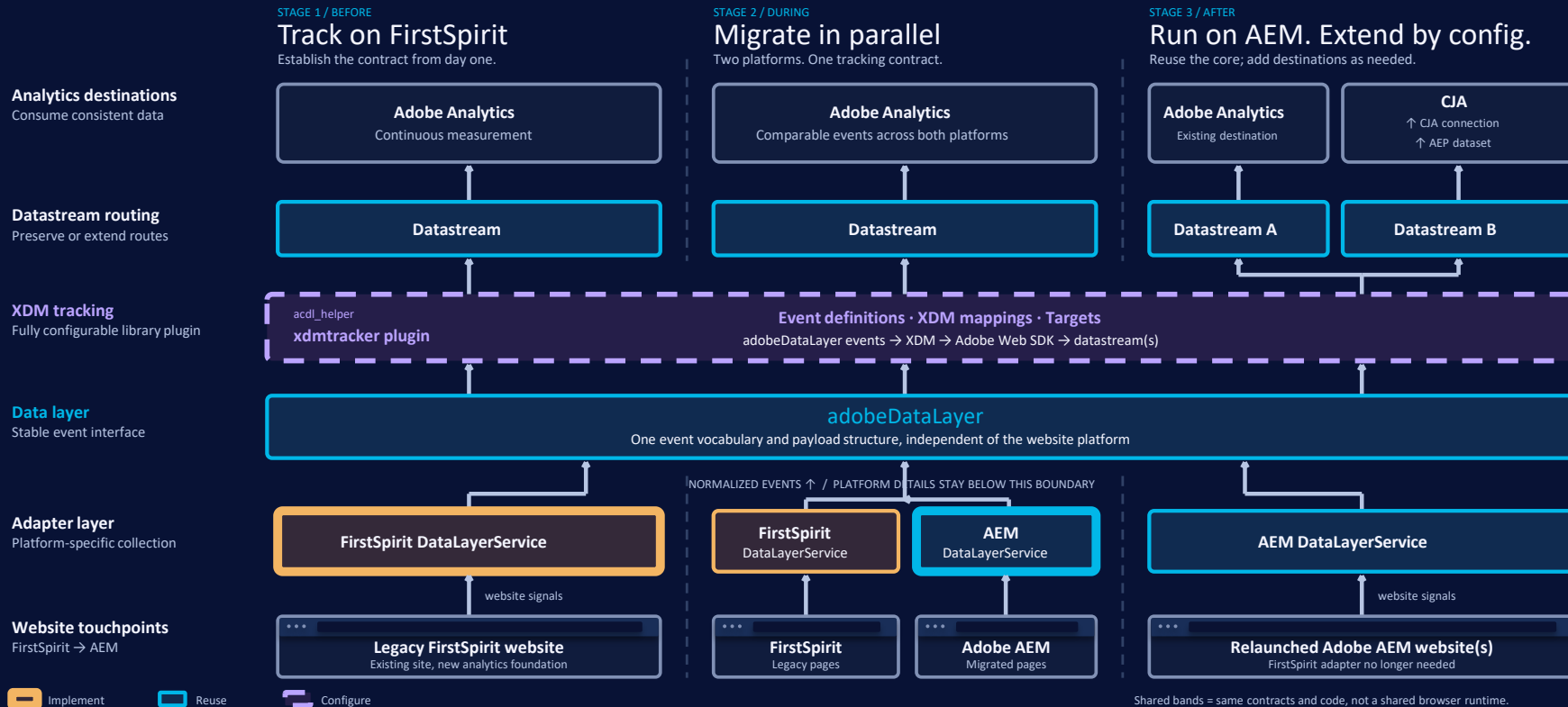
Multiple
Data Views

Country A

Contact Form

Application A, B, C

Technical: Modular architecture



Track on FirstSpirit

Establish the tracking contract before the platform changes.

STAGE 1 / BEFORE

Implement the adapter

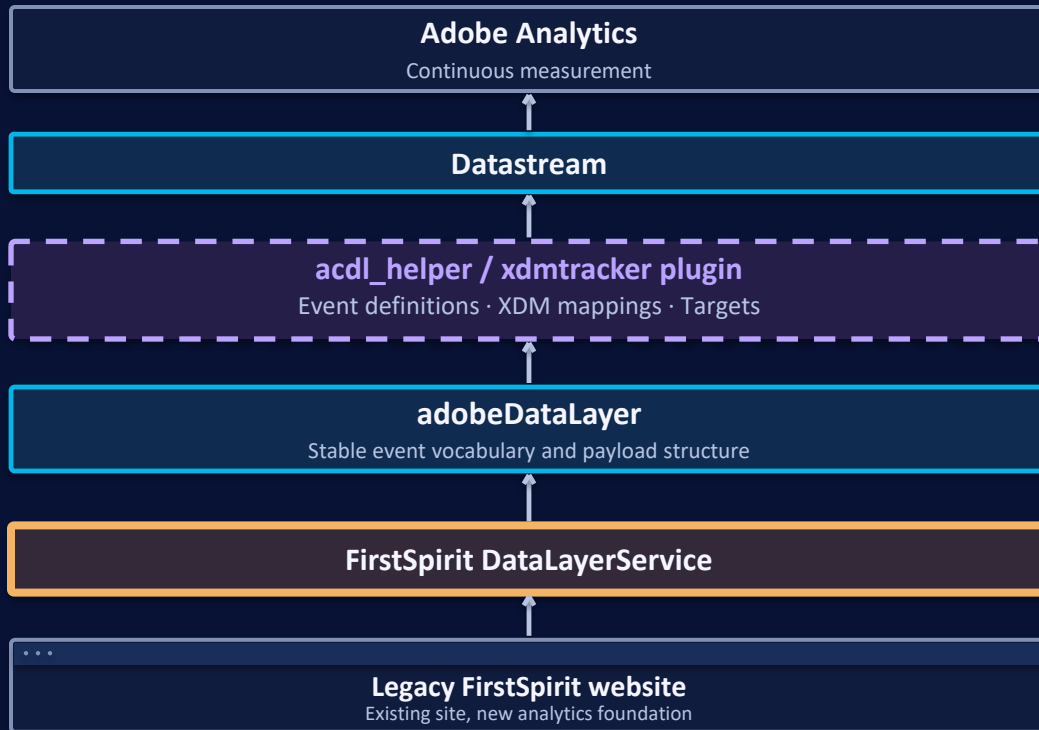
Translate FirstSpirit website signals into normalized data-layer events.

Configure the definitions

Define events, XDM mappings and targets in the xdmtracker plugin.

Keep the destination

Route the payload to Adobe Analytics through the datastream.



Migrate in parallel

Two website platforms. One tracking contract.

STAGE 2 / DURING

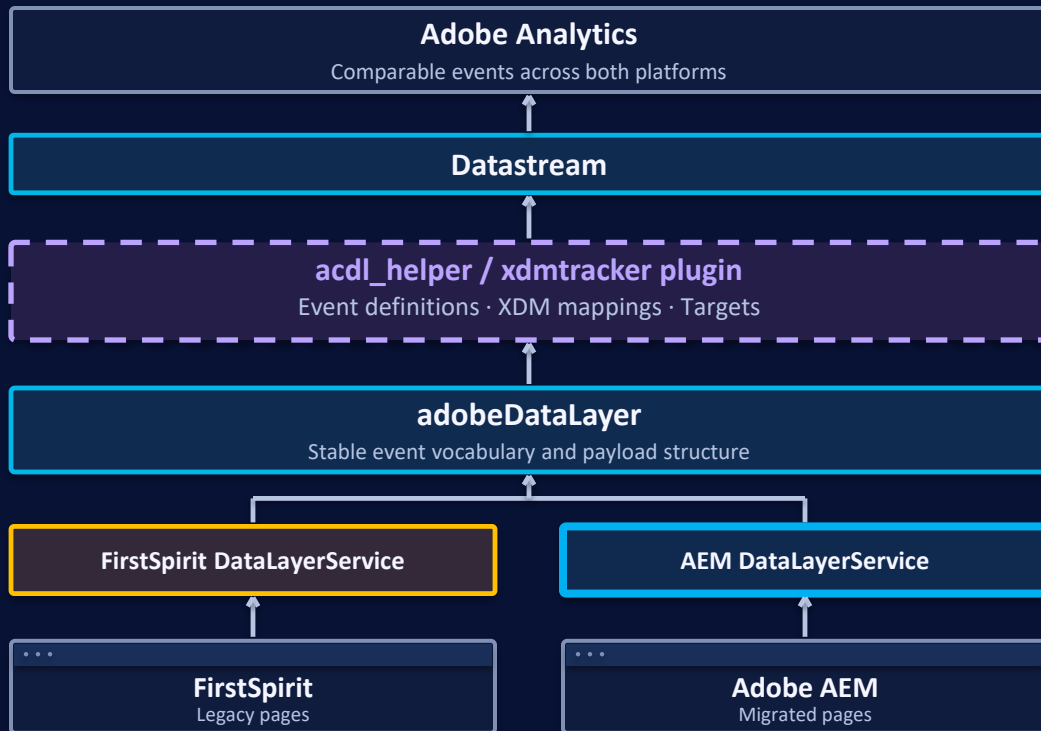
Implement platform-specific collection for the migrated AEM pages.

Reuse the shared contract

Keep the event vocabulary, payload structure and tracker definitions consistent.

Measure both platforms

Use the same datastream and Adobe Analytics destination for comparable events.



Run on AEM. Extend by config.

Reuse the core; add destinations as needed.

STAGE 3 / AFTER

Retire the legacy adapter

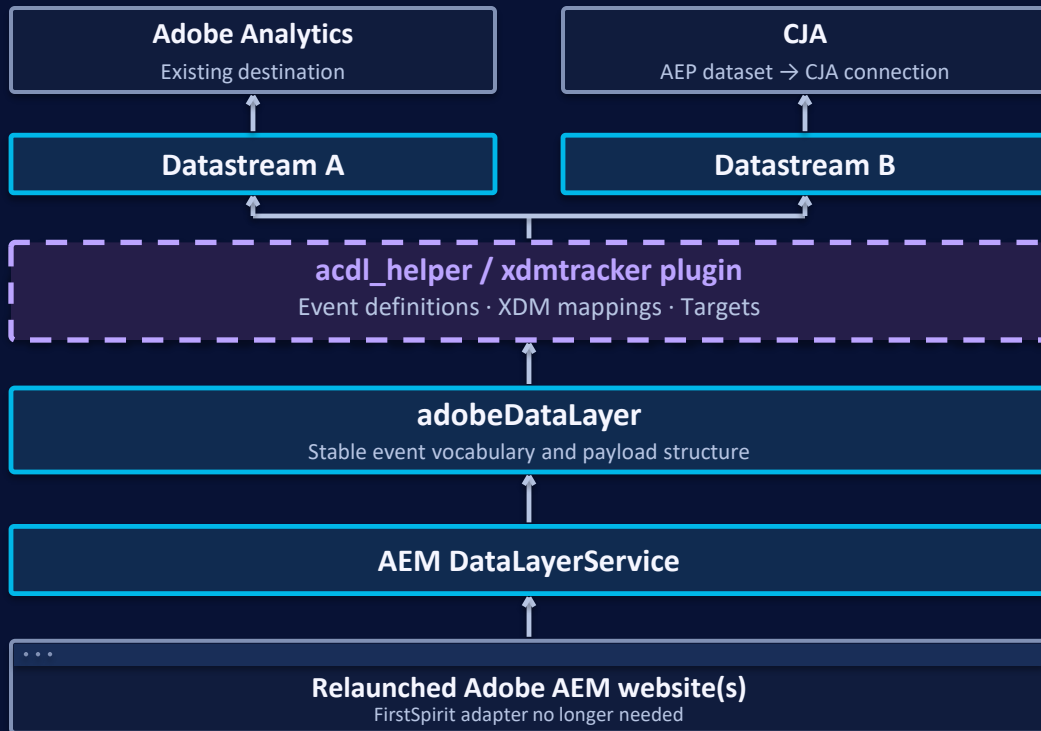
Run the relaunched websites on AEM; the FirstSpirit adapter is no longer needed.

Retain the tracking core

Reuse `adobeDataLayer`, the AEM adapter and the configurable `xdmtracker` plugin.

Extend the destinations

Keep Adobe Analytics and configure an additional route through an AEP dataset to CJA.



THANK YOU

