



Creating Headless Omnichannel Experiences with Adobe Experience Manager 6.5

Amol Anand



1

Use one CMS to **manage content** across channels

2

Prepare content for both **traditional and headless** delivery

3

Reuse and publish once to multiple channels

about me



Amol Anand

Principal Consultant

amol@adobe.com



/amol-anand

GitHub

@iamolanand

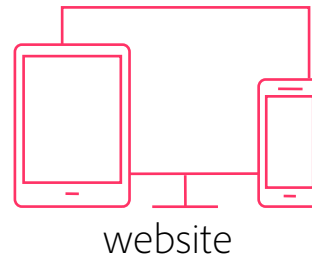
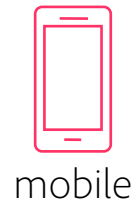
twitter

/amol-anand

LinkedIn

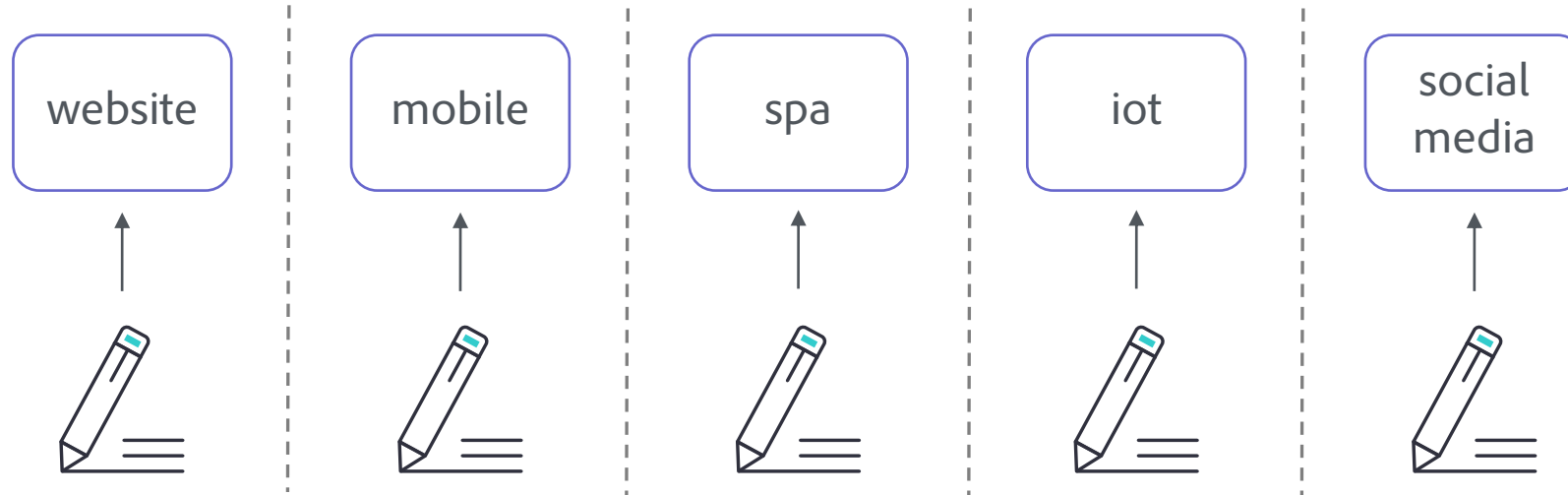
use one CMS to **manage content**
across channels

explosion of digital channels



Challenge | How can I support these different technologies?

independent systems



authoring content with different systems for different channels

-
- message consistency
 - duplication of effort
 - content silos
 - increased costs

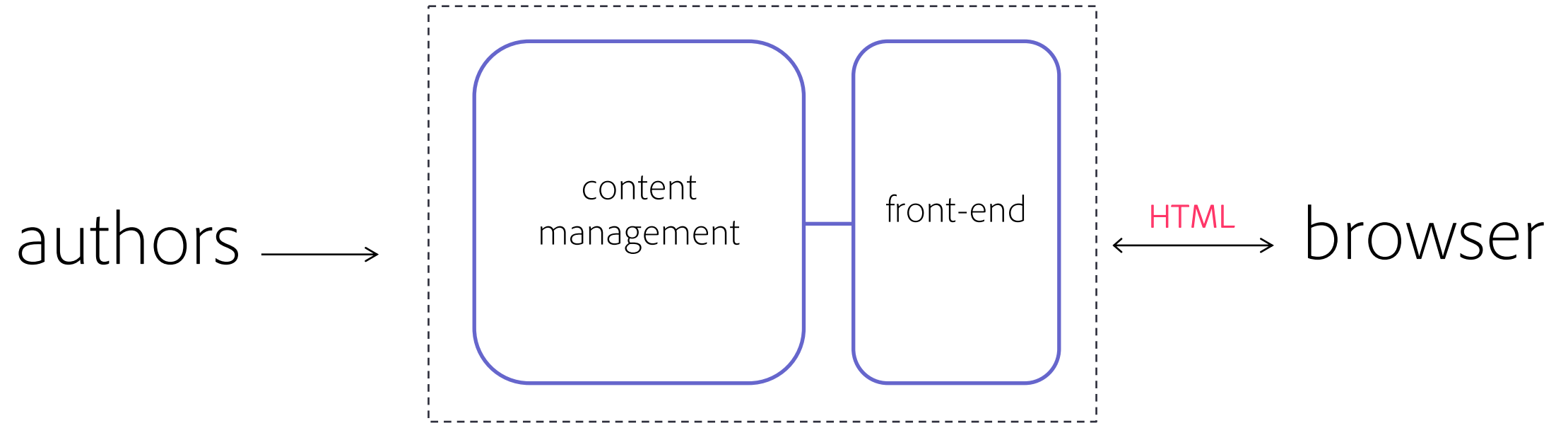
one system | modular content



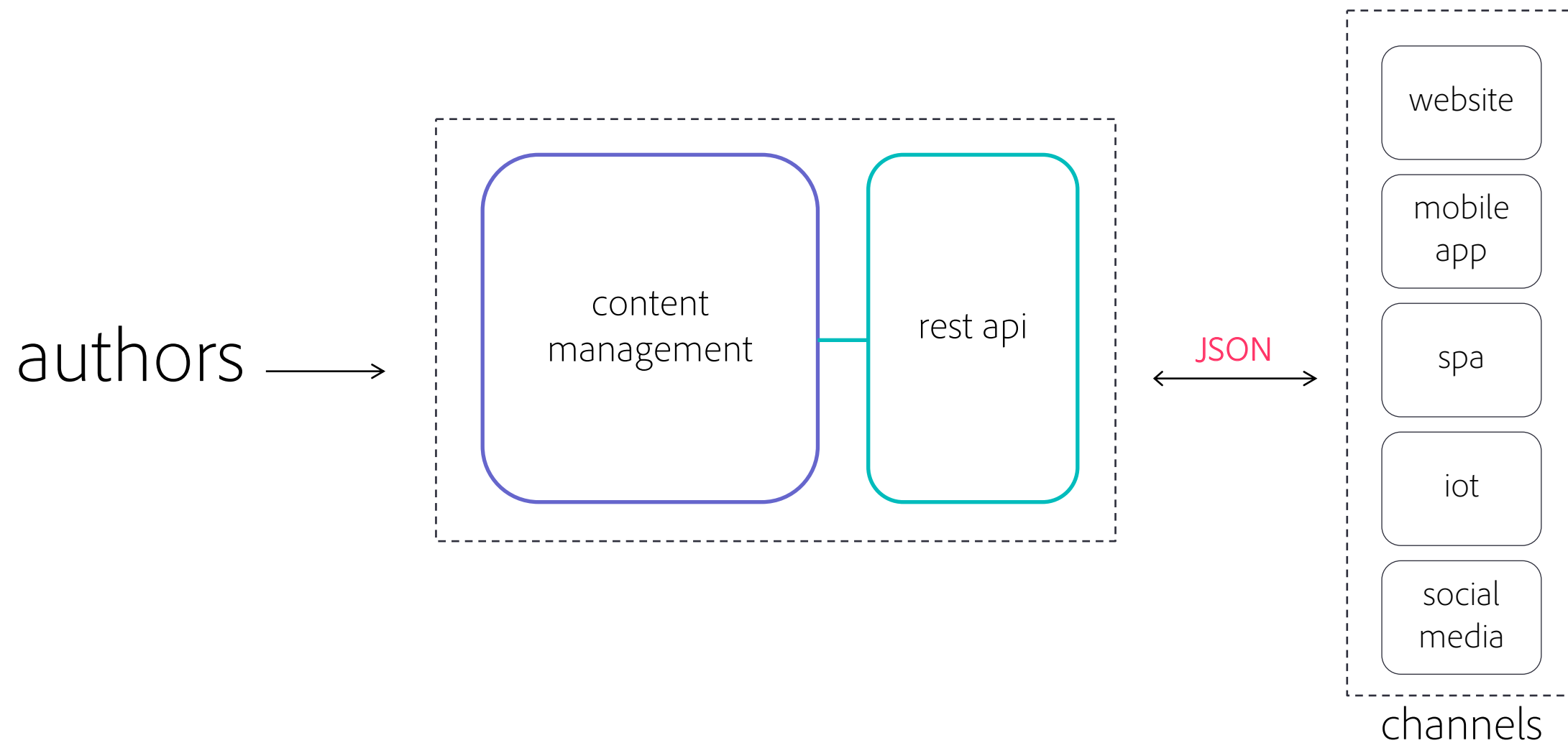
modular content

content and experience
fragments

traditional cms



headless cms



traditional vs headless

traditional

marketers & developers

fully formatted content

single application stack,
end-to-end control

multi-channel support



headless

developers

raw content via API

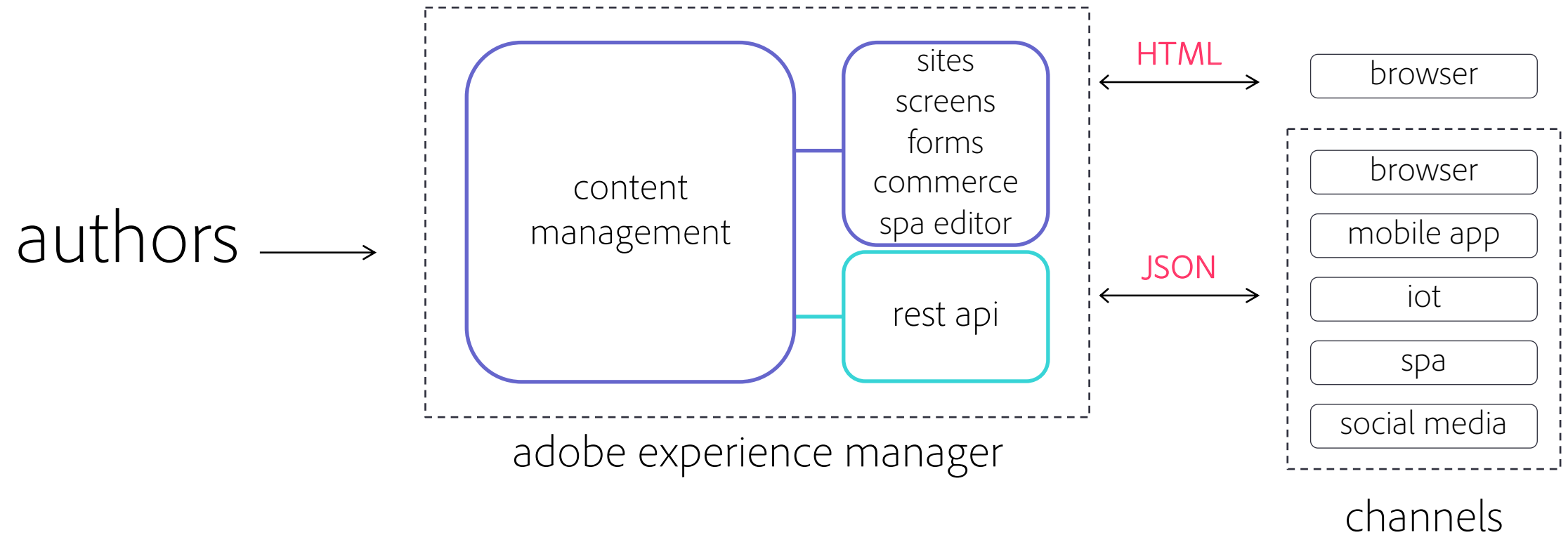
multiple downstream technologies,
scalable

additional application stacks



why choose? you will need both!

'hybrid' cms = traditional + headless



Prepare content for both
traditional and headless delivery

“headless”

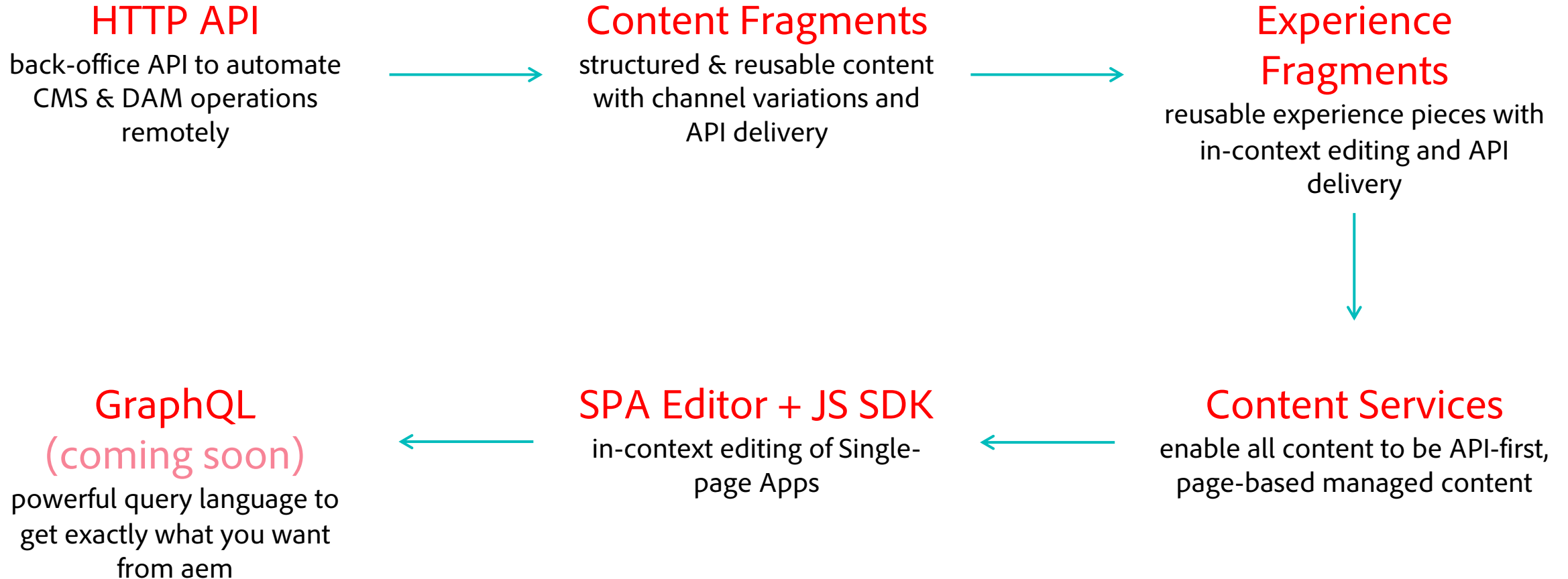
headless delivery = cms delivering the content as JSON, instead of HTML.

headless editing = author editing the content out of the context where it will be used.

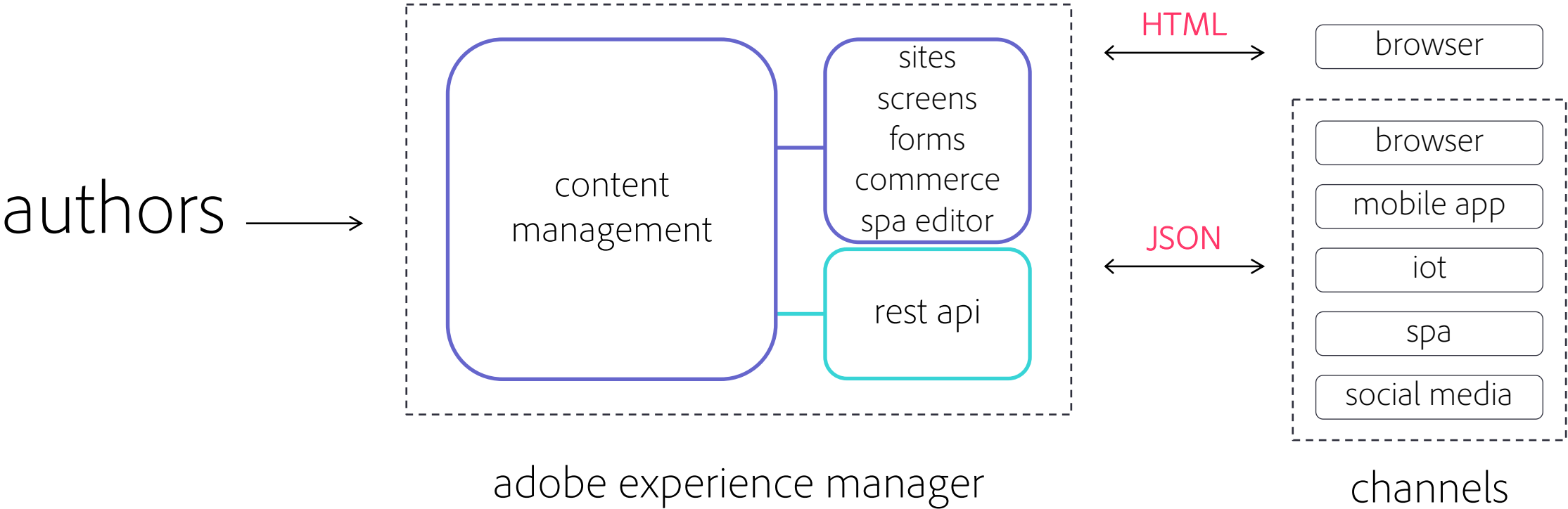
headless
authoring \neq headless
delivery

headless delivery does not have to imply headless editing

key aem capabilities | omnichannel



hybrid cms



headless delivery | JSON

Content Services

Assets HTTP API

Custom Sling Model Exporter

- OOTB product feature
- faster time to value
- UI driven
- “managed” JSON
- custom components need sling models

headless delivery | JSON

Content Services

Assets HTTP API

Custom Sling Model Exporter

- OOTB product feature
- faster time to value
- exposed automatically (CRUD)
- supports content fragments
- less control over JSON
- tightly coupled to DAM structure

headless delivery | JSON

Content Services

Assets HTTP API

Custom Sling Model Exporter

- full control over JSON schema
- add business logic and dynamic behavior
- reuse existing non-modular content
- requires development
- customer maintained

delivering omnichannel experiences: personas

1. Content Architect



Content Model

2. Content Author



Content Fragment

3. AEM Developer



Component

HTML



{API}



JSON

4. AEM Web Producer



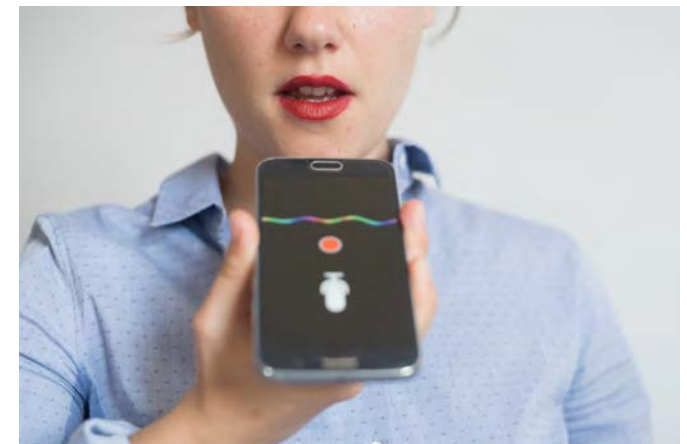
Web Page

5. Developer



External Applications

increase customer reach



Reuse and publish once to multiple channels

(single page app, mobile, voice, chatbot, and email)

delivering to each channel

website

spa

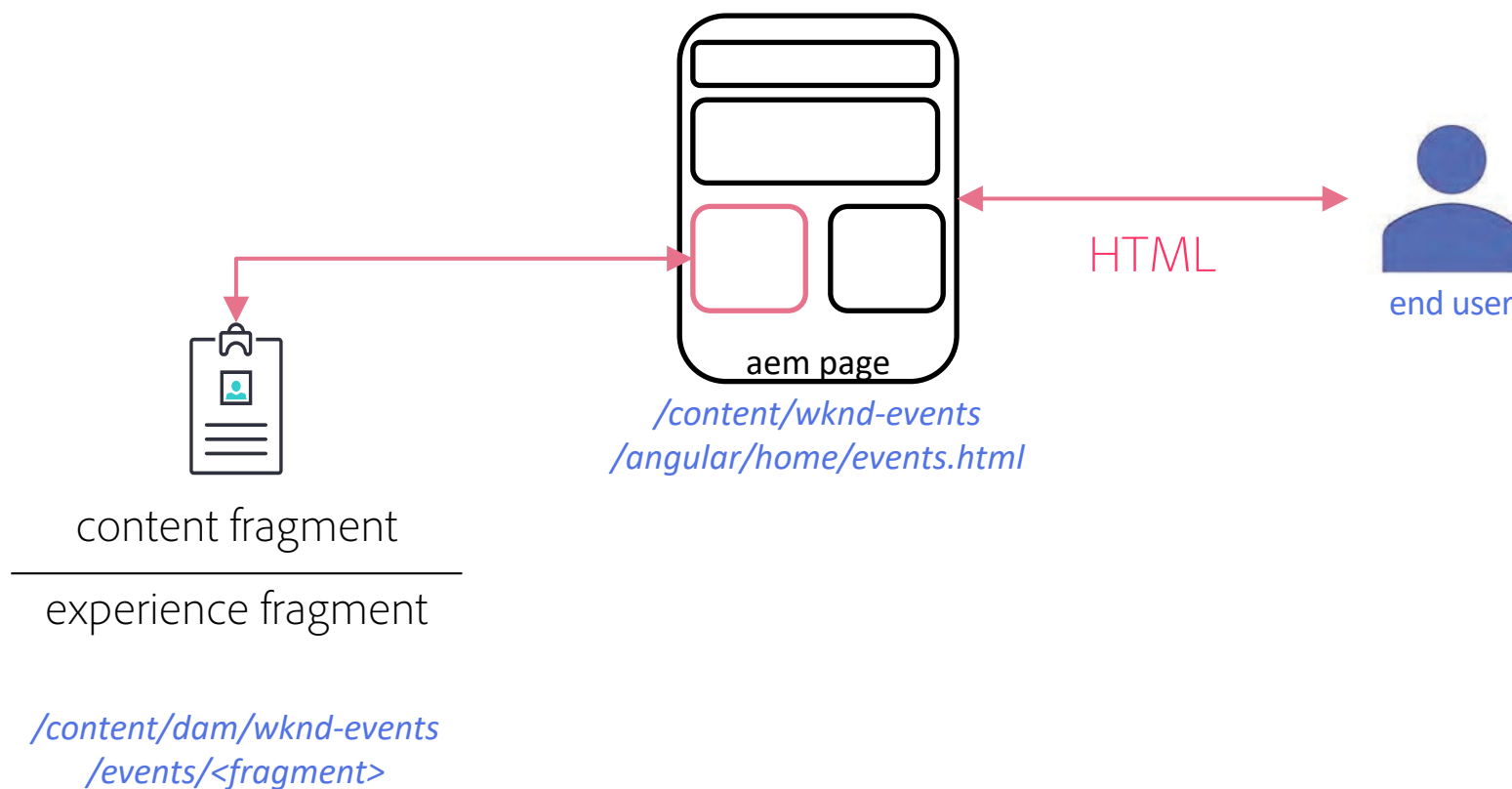
mobile

email

voice

chat

Drag & Drop fragment onto AEM page and publish



delivering to each channel

website

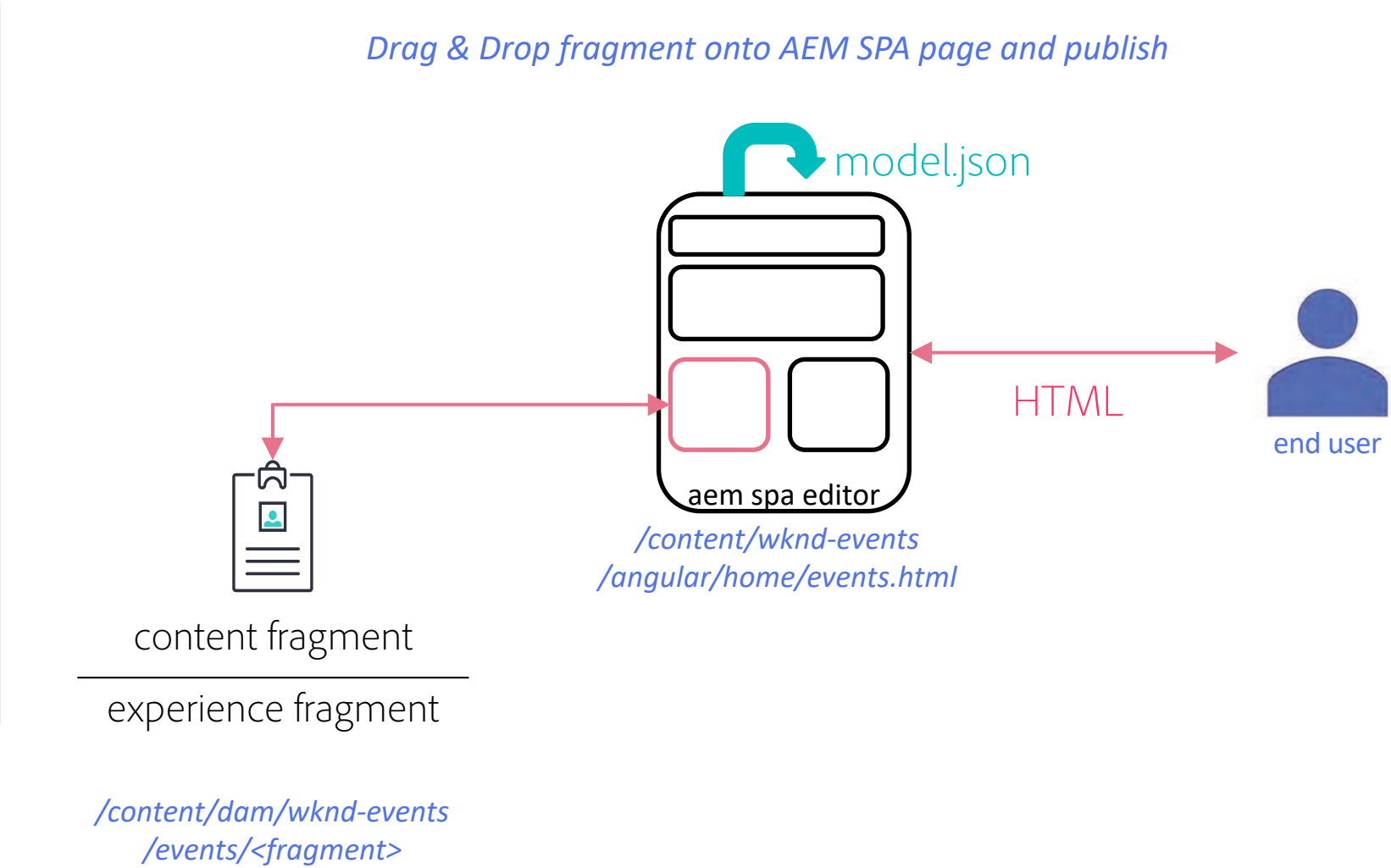
spa

mobile

email

voice

chat



delivering to each channel

website

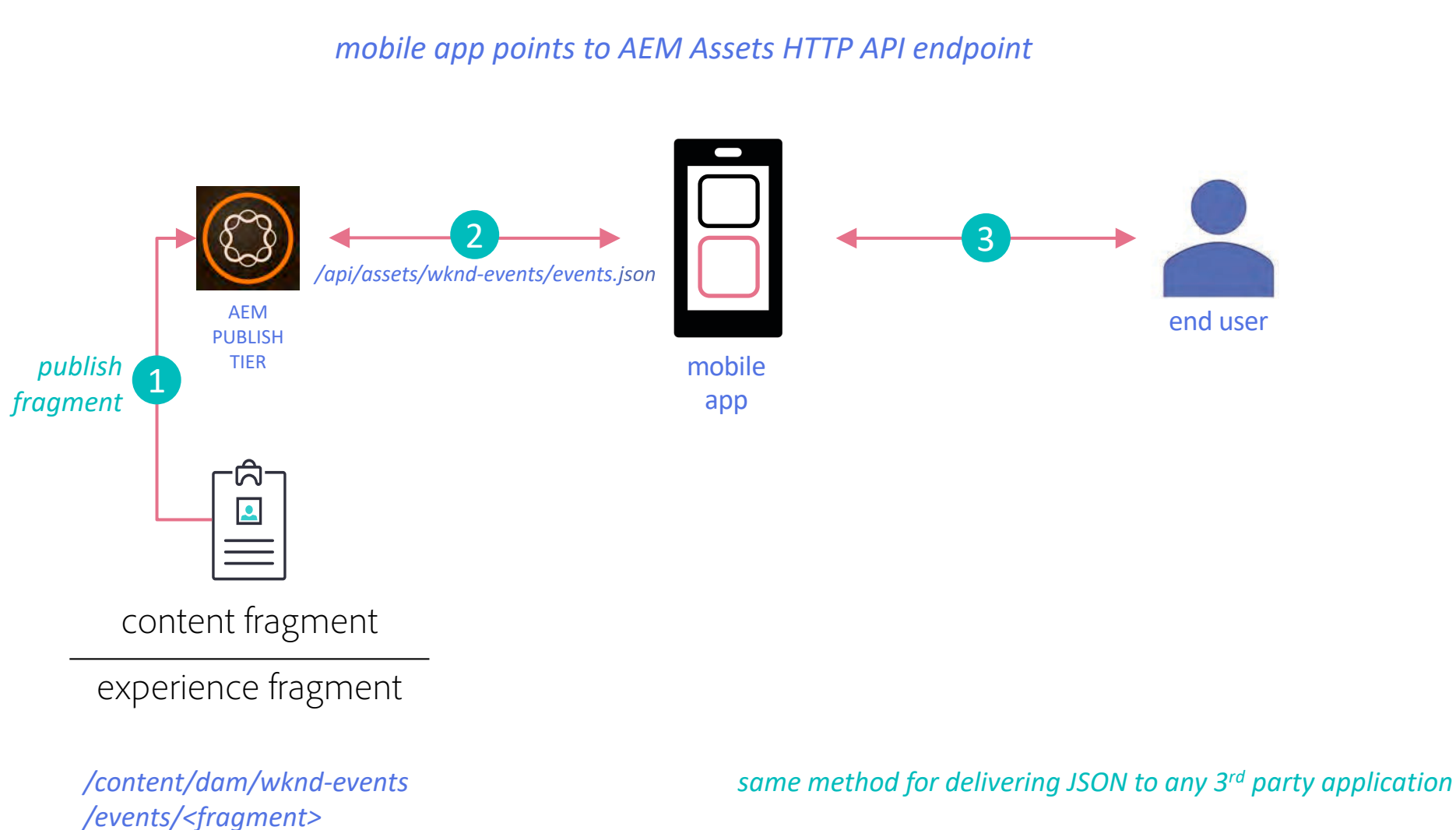
spa

mobile

email

voice

chat



delivering to each channel

website

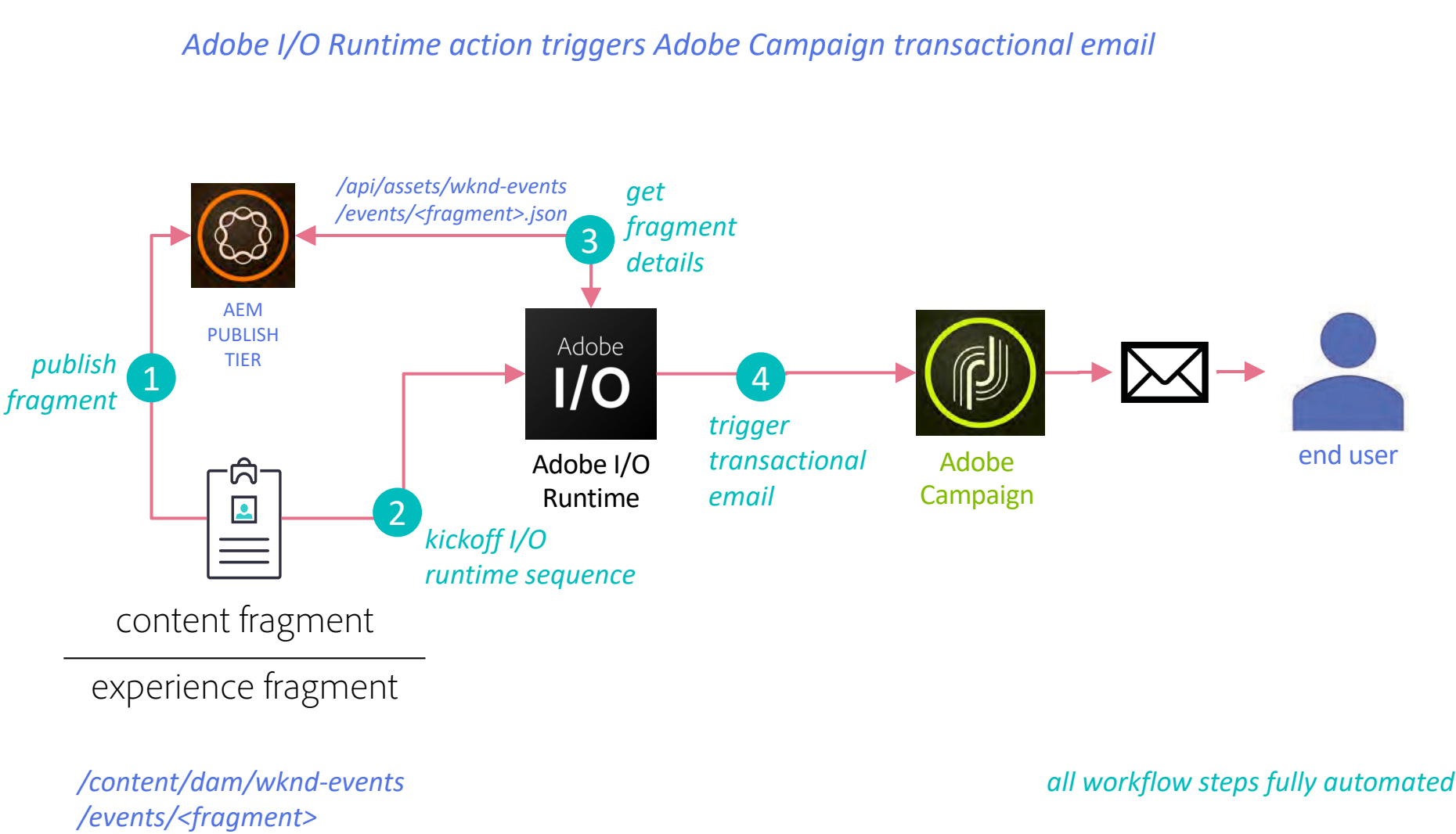
spa

mobile

email

voice

chat



delivering to each channel

website

spa

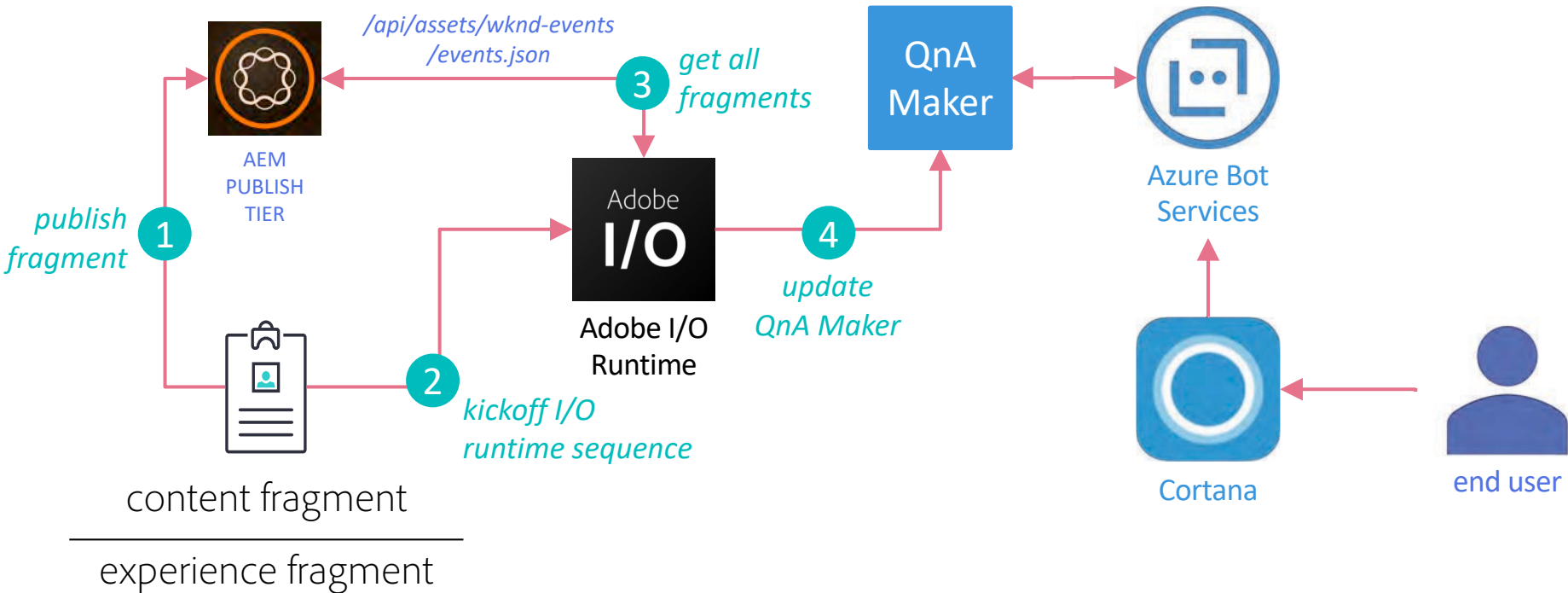
mobile

email

voice

chat

Adobe I/O Runtime action feeds AEM content to Microsoft QnA Maker and Azure Bot Services



`/content/dam/wknd-events
/events/<fragment>`

all workflow steps fully automated

delivering to each channel

website

spa

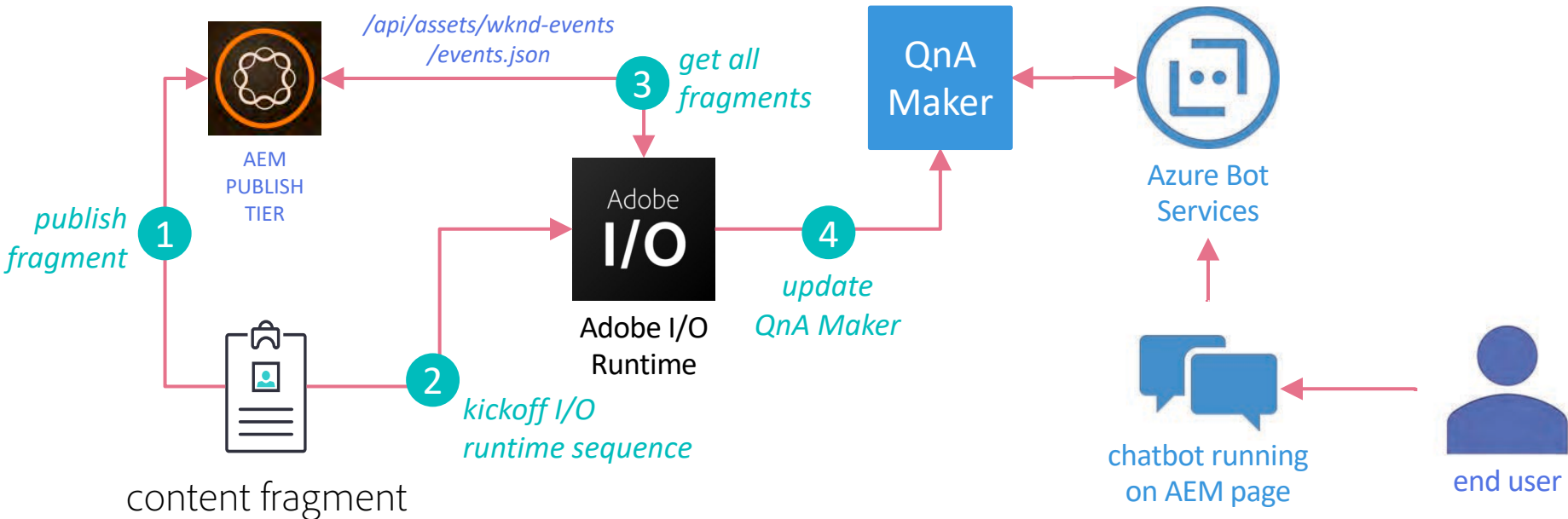
mobile

email

voice

chat

Adobe I/O Runtime action feeds AEM content to Microsoft QnA Maker and Azure Bot Services



`/content/dam/wknd-events`
`/events/<fragment>`

all workflow steps fully automated

1

Use one CMS to **manage content** across channels

2

Prepare content for both **traditional and headless** delivery

3

Reuse and publish once to multiple channels

Q & A

Amol Anand

amol@adobe.com

Twitter: **@iamolanand**

LinkedIn: **/in/amolanand**



Adobe



Adobe