API Working Group

September 26, 2019

Includes notes from meeting

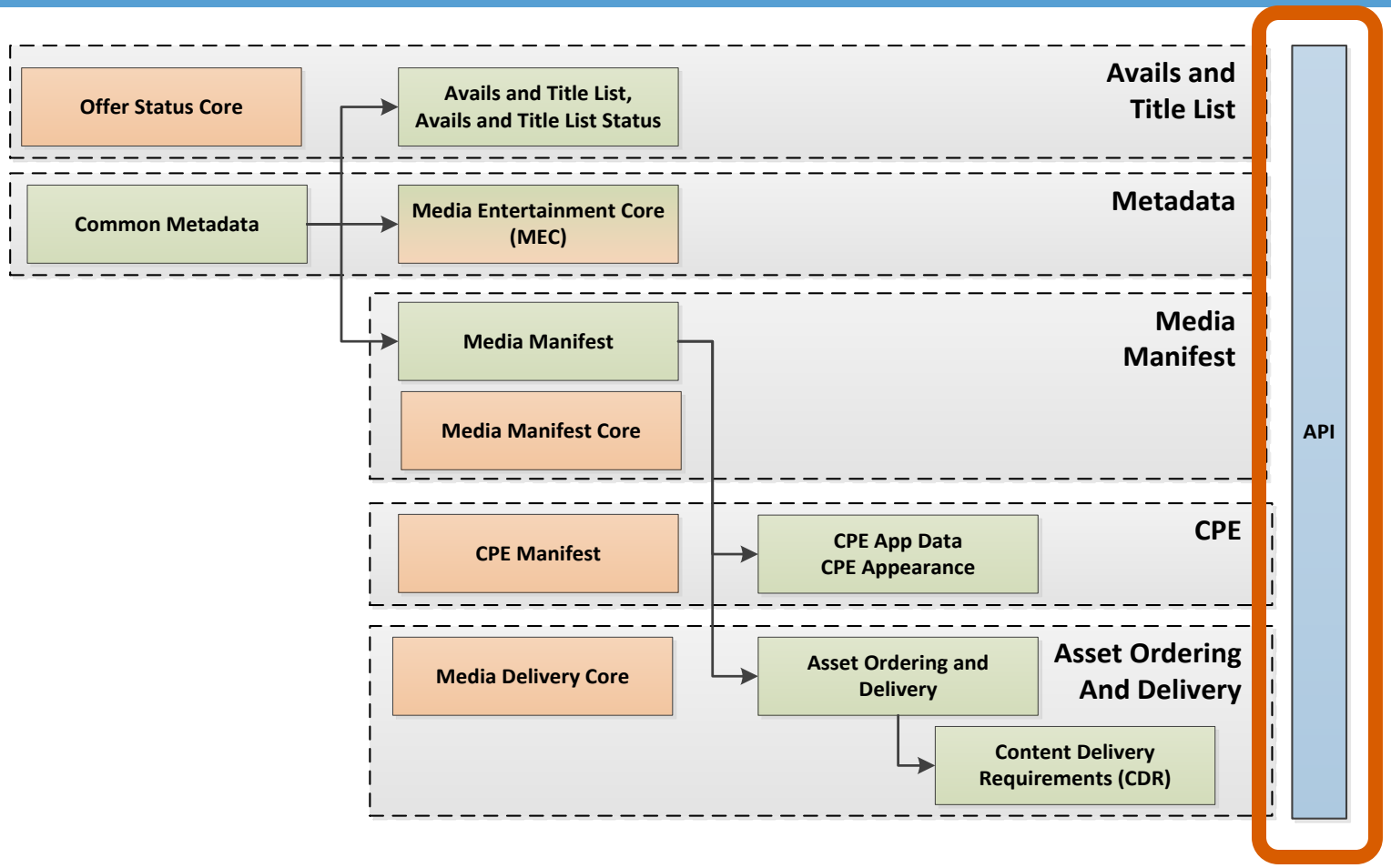
Agenda

- API Spec Updates based on last meeting
- Asset Ordering Spec
- MovieLabs' prototype progress (OpenAPI 3.0)
- Discussion and Next Steps

Spec Updates

API Spec Updates

- Updated spec posted after last meeting
  - www.movielabs.com/md/api
- Biggest change is removal of content provider from URL



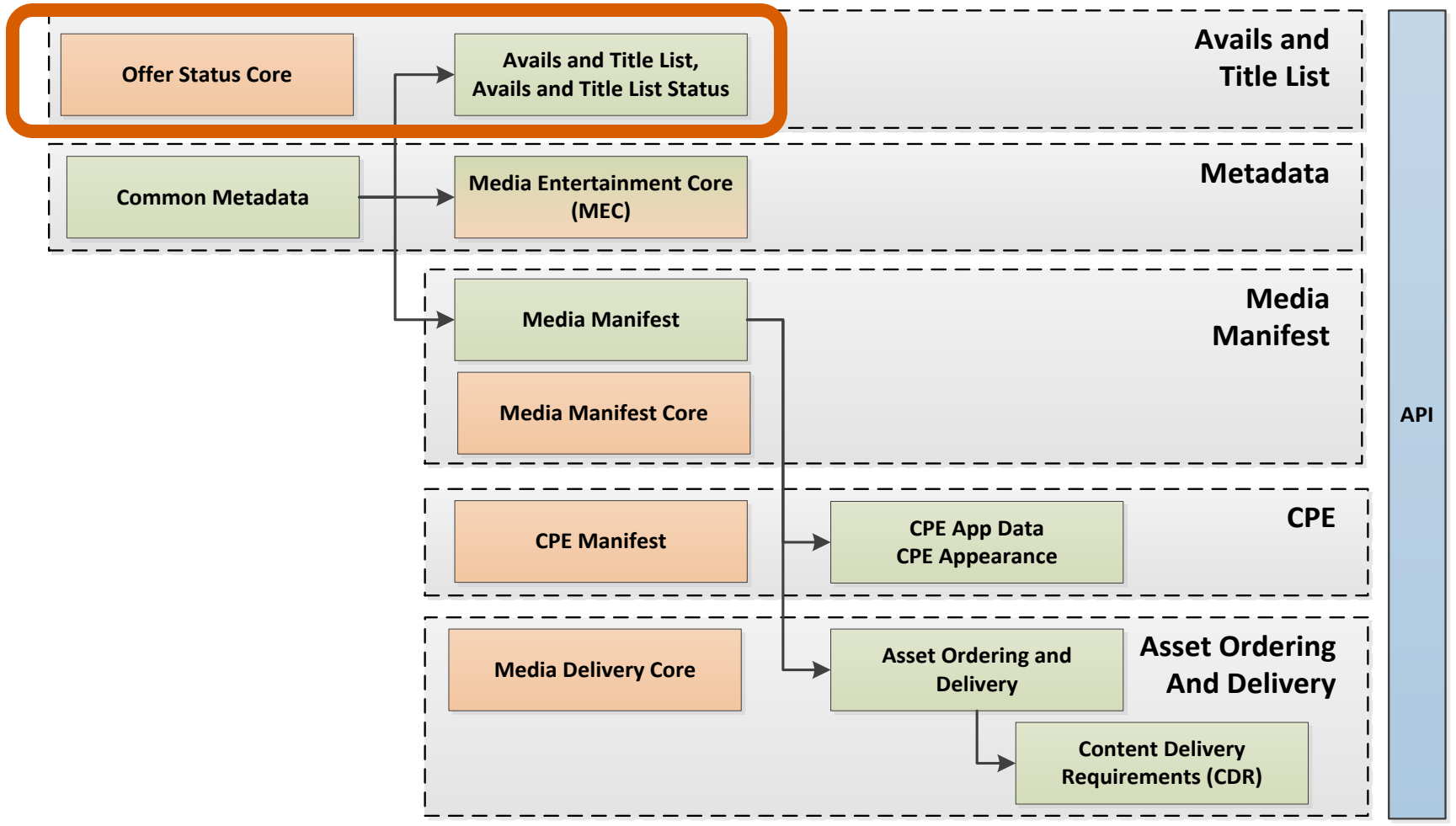
URL

- **Tentative Conclusion:**
  - Rather than having studio in path, use query parameters.
  - They would not all necessarily be required
    - Studio
    - ALID
    - License Model (LicenseType?)
    - Territory
- …/{ALID}?licensor=sofaspud&model=TVOD
- …/{ALID}/{TransactionID}?licensor=…

OfferStatus

Offers Status Specs

- OfferStatus added to Avails
- New “Offer Status Core” Spec written to define basic use cases, to demystify OfferStatus and to provide examples

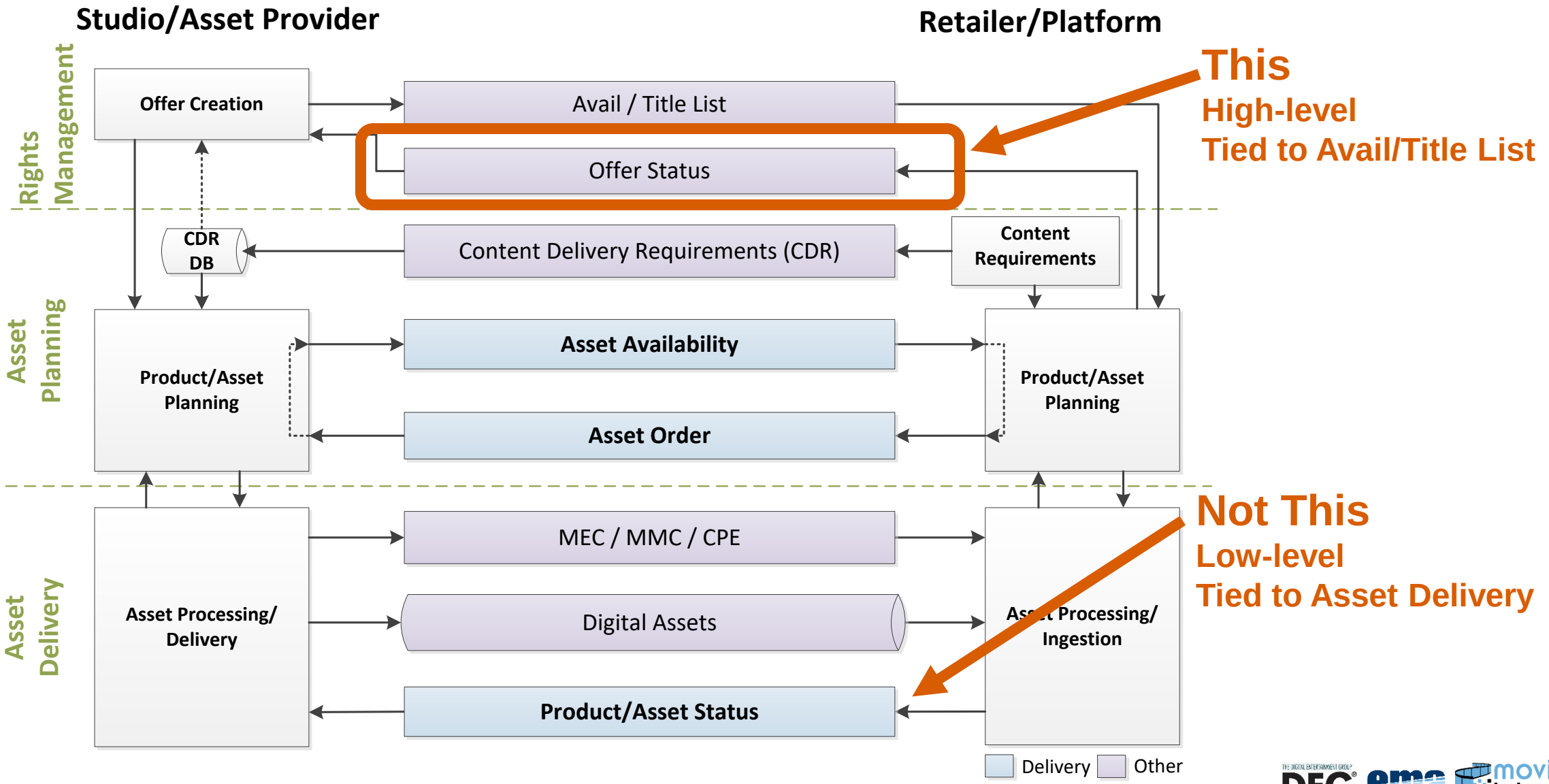


Offer Status

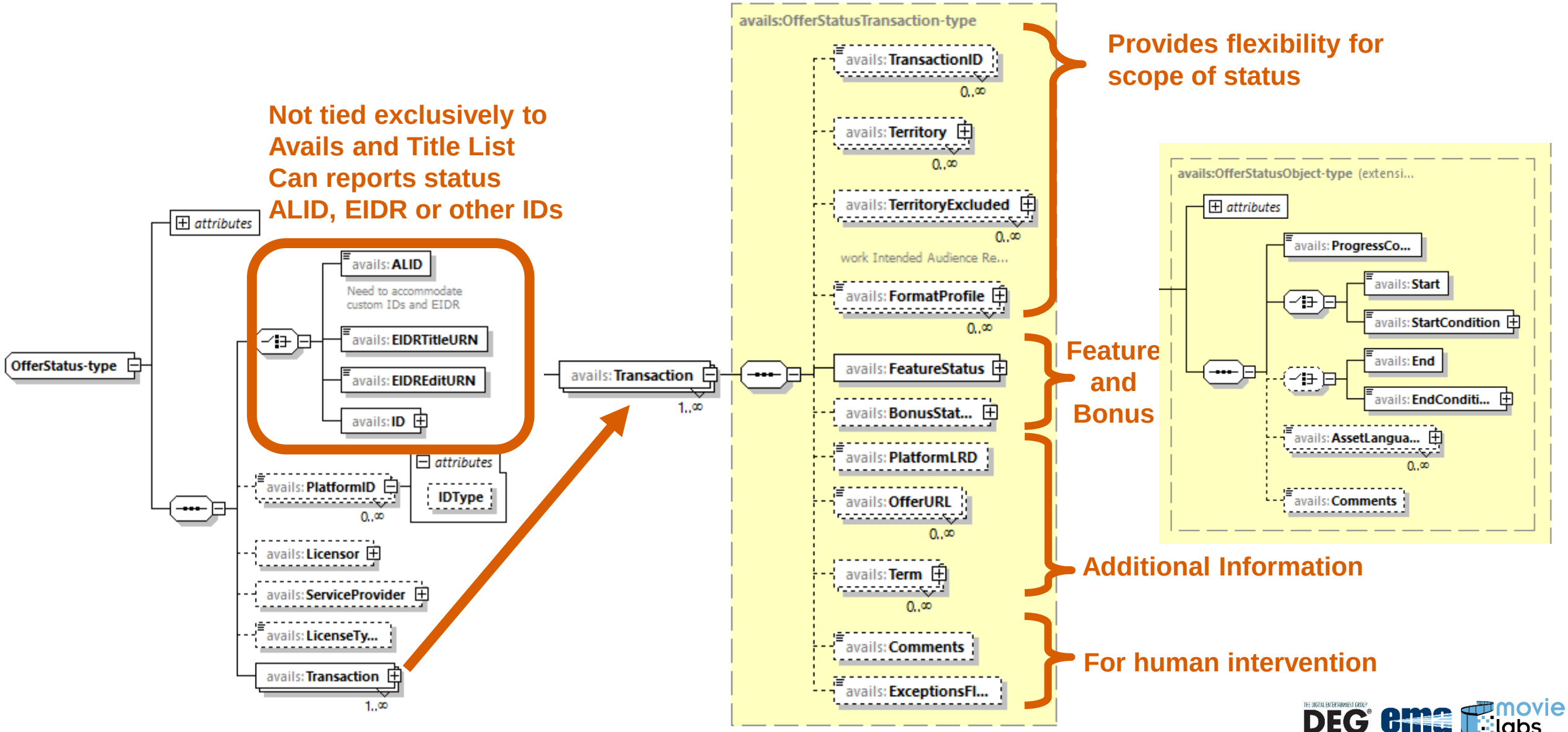
- Offer Status designed to provide status on Avail or Title List

- Primary goal is to determine if there are disconnects, before title goes live (early, late, wrong price, etc.)
- Echoes Avail or Title List terms, including refinements made by platform
  - For example, could be subset of languages, or actual price
- High-level status: Live, Ready, Blocked, In-Process

Offer Status in context

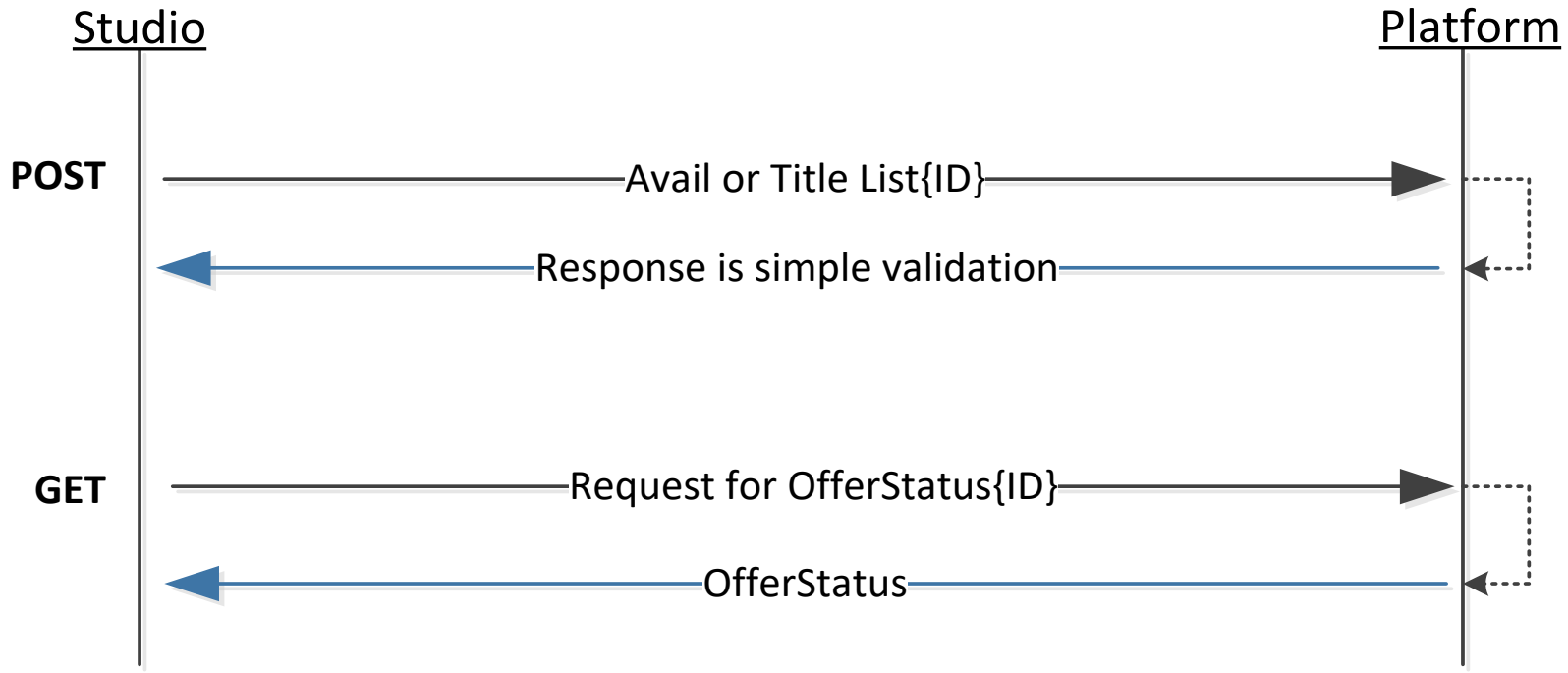


XML

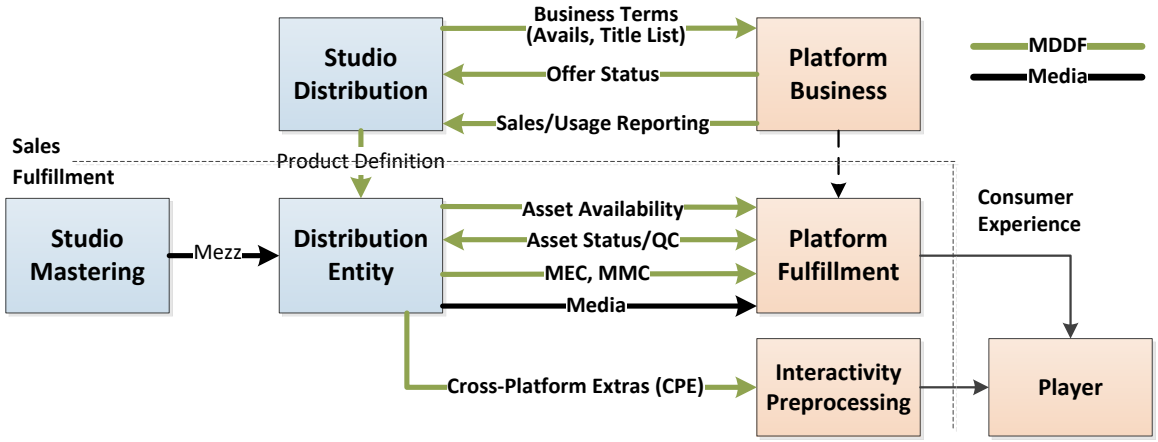

Not tied exclusively to
Avails and Title List
Can reports status
ALID, EIDR or other IDs
OfferStatus-type
attributes
avails: ALID
Need to accommodate
custom IDs and EIDR
avails: EIDRTitleURN
avails: EIDREditURN
avails: ID
avails: PlatformID
0..∞
attributes
IDType
avails: Licensor
avails: ServiceProvider
avails: LicenseTy...
avails: Transaction
1..∞
avails: Transaction
1..∞
avails:OfferStatusTransaction-type
avails: TransactionID
0..∞
avails: Territory
0..∞
avails: TerritoryExcluded
0..∞
work Intended Audience Re...
avails: FormatProfile
0..∞
avails: FeatureStatus
avails: BonusStat...
avails: PlatformLRD
avails: OfferURL
0..∞
avails: Term
0..∞
avails: Comments
avails: ExceptionsFl...
Provides flexibility for
scope of status
Feature
and
Bonus
Additional Information
For human intervention
avails:OfferStatusObject-type (extensi...
attributes
avails: ProgressCo...
avails: Start
avails: StartCondition
avails: End
avails: EndConditi...
avails: AssetLangua...
0..∞
avails: Comments

OfferStatus in API

- Assume initial return status is simple validation (*not* OfferStatus)
- OfferStatus provides detailed status on offer that will update over time
  - Data are time tagged to indicate when status was taken



API Prototyping



Supply Chain API Planning

Goal

Develop standard API to enable fully automated delivery of MDDF Avails, Media Entertainment Core and Media Manifest Core

Approach

Based on existing draft specification

- Work with industry partners to develop working prototype
- Refine and finalize route specifications
- Refine and finalize security and authentication standards

Use open standards to describe API ensuring consistent deployment across multiple parties

Open API 3.0

- Standardized way of describing REST API's, now managed by Linux Foundation (formerly referred to as Swagger)
- Allows for multitude of tools to ease implementation of REST API's including the popular toolset Swagger
- Specify routes, parameters, responses, etc. using JSON/YAML
- Documentation and HTML test pages can be generated automatically with open tools for multiple development environments
- Stubs are created to hook application code to
- Same spec can therefore be deployed across multiple environment

Open API 3.0 Example

Example YAML for GET route

```yaml
paths:
  '/mec/{contentId}':
    get:
      tags:
        - MEC Operations
      description: "Returns an MEC resource to the requester"
      operationId: mecGetResource
      parameters:
        - $ref: '#/components/parameters/mecContentId'
      responses:
        '200':
          description: "Successful"
          content:
            application/xml:
              schema:
                $ref: "#/components/schemas/mecResource"
        '404':
          description: "Failed: Resource not found"
          content:
            application/xml:
              schema:
                $ref: "#/components/schemas/error"
```

Define route path and parameters

Stub for hooking application code to

Define response, success and failure

Reusable components can be defined for response, input parameters, etc.

Example Node code

Stubs (operationIds) allow easy integration into code base

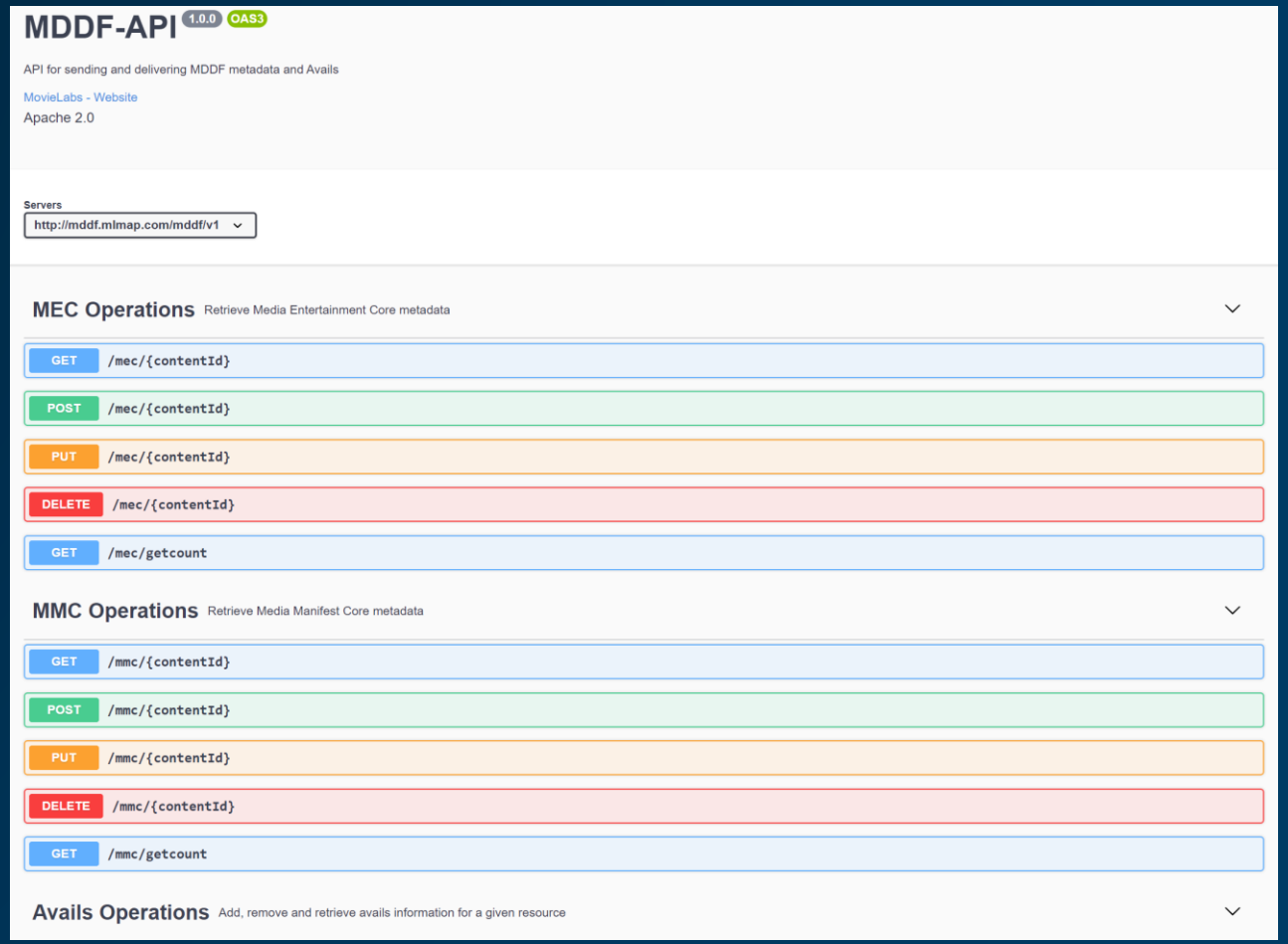
```
module.exports = async function swagger(dependencies) {
    const { expressApp, database } = dependencies;
    const apiDoc = path.join(__dirname, '../api/docs-v1/mddf-api-doc.yaml');
    const swaggerDocument = yaml.load(apiDoc);
    expressApp.use('/api-docs', swaggerUi.serve,
swaggerUi.setup(swaggerDocument));

    initialize({
        app: expressApp,
        apiDoc,
        dependencies: {
            database,
        },
        operations: {
            mecGetCount: await mecController.mecGetResource,
            mecGetResource: await mecController.mecGetResource,
            mecPostResource: await mecController.mecPostResource,
            mecPutResource: await mecController.mecPutResource,
            mecDeleteResource: await mecController.mecDeleteResource,
            mmcGetCount: await mmcController.mmcGetResource,
            mmcGetResource: await mmcController.mmcGetResource,
            mmcPostResource: await mmcController.mmcPostResource,
            mmcPutResource: await mmcController.mmcPutResource,
            mmcDeleteResource: await mmcController.mmcDeleteResource,
            artworkGetResource: await artworkController.artGetResource,
            mecMapResource: await mecMapController.mecMapResource,
            searchTitles: await searchController.getSearch,
        },
    });
```

Website

http://mddf.mlmap.com/api-docs/

Swagger can create a functioning web based front end for testing, providing documentation and examples



MovieLabs Prototype

- Github Repo has first Node JS prototype of basic routes allowing GET, POST, PUT & DELETE of resources

  Code base in Node JS to simulate backend database

- Includes the Open API 3.0 specification
- Code base in Node JS to simulate backend database
- Provides a starting point for an implementation and further testing and discussion of remaining open issues

Resources

Open API 3.0: https://www.openapis.org

Open API Tools: https://openapi.tools

Swagger tools: https://swagger.io/

Github Repo: https://github.com/MovieLabs/MDDF_API

Example: mddf.movielabs.com/api-docs

Discussion

- Any reason not to use OpenAPI 3.0?
  - Should this be the official documentation for this part of the API?
- Our goal is Avails, but MEC is a much easier test subject. Should we test with MEC?

Any Other Business?

THE DIGITAL ENTERTAINMENT GROUP DEG®

ema

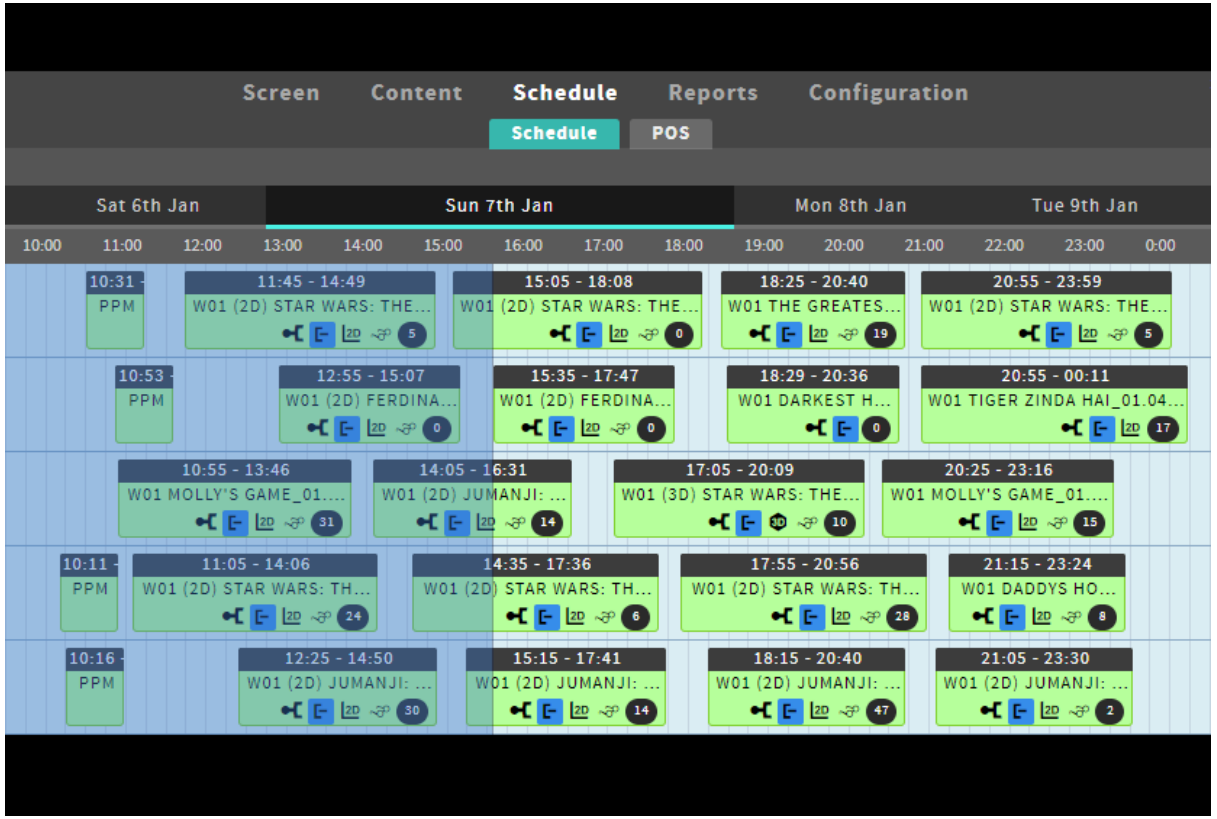
movie labs

Wrap-up

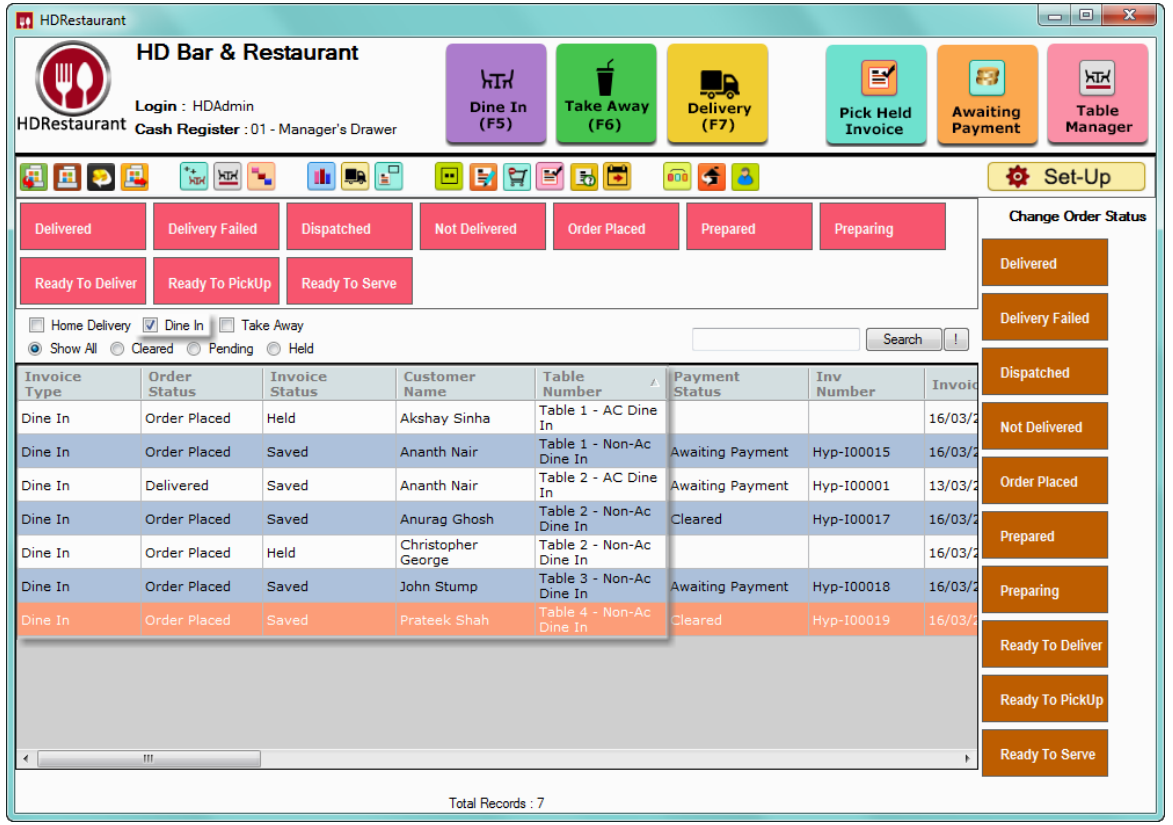
- Action Items
  - Focus on Avails Status (no current solution, easy to motivate stakeholders)
  - OpenAPI 3.0 with Avails and OfferStatus will be a major step towards adoption
    - ~~Could potentially use OfferStatus with Excel Avail?~~
- Next Steps
  - Next meeting?

BACKUP

Dashboards…



Arts Alliance Media



HD Restaurant