

# Office of Justice Programs

---

## OJP OCIO Solution Patterns Definitions *DRAFT*



**Innovation • Partnerships  
Safer Neighborhoods**

**Version 1.4.3**

**Last Updated:** June 14, 2017

**Prepared for:**

Department of Justice  
Office of Justice Programs  
810 7<sup>th</sup> Street, NW Room 8126  
Washington, DC 20531

**Prepared by:**

Ira Baron, CTO, OJP OCIO



## Table of Contents

<b>1</b>	<b>OVERVIEW.....</b>	<b>4</b>
1.1	Context.....	4
1.2	Purpose.....	4
1.3	Audience.....	4
1.4	Scope.....	5
<b>2</b>	<b>BUSINESS SOLUTIONS .....</b>	<b>6</b>
<b>3</b>	<b>CAPABILITY PATTERNS .....</b>	<b>7</b>
3.1	Approval Workflow.....	7
3.2	Case Management.....	7
3.3	Content Management.....	7
3.4	Curriculum Management.....	8
3.5	Customer Engagement.....	8
3.6	Dashboard View.....	8
3.7	Data Analysis.....	8
3.8	Decision Support.....	9
3.9	Document Assembly.....	9
3.10	eCommerce.....	9
3.11	Editorial Workflow.....	9
3.12	Information Publication.....	9
3.13	Video Content Management.....	10
3.14	Visualization.....	10
3.15	Web Conferencing.....	10
3.16	Input and Output.....	10
	3.16.1 Collection.....	10
	3.16.2 Dissemination.....	10
<b>4</b>	<b>FEATURE PATTERNS.....</b>	<b>11</b>
4.1	Access Control.....	11
4.2	Browser Access.....	11
4.3	Computation and Analytics.....	11
4.4	Correspondence Management.....	11
4.5	Data Collection.....	11
4.6	Data Management.....	11
4.7	Document Collection.....	12
4.8	Document Management.....	12
4.9	Identity Management.....	12
4.10	Reporting.....	12
4.11	Search.....	12
4.12	Service Access.....	13
4.13	Search.....	13
4.14	Tracking.....	13
4.15	Workflow.....	13

**5 IMPLEMENTATION PATTERNS..... 14**

- 5.1 API ..... 14
- 5.2 API-Gateway ..... 14
- 5.3 Authentication..... 14
- 5.4 Authorization ..... 14
- 5.5 Content Caching ..... 14
- 5.6 Content Distribution..... 14
- 5.7 Data Storage ..... 14
- 5.8 Document Storage ..... 14
- 5.9 Email ..... 14
- 5.10 Logging..... 15
- 5.11 Resource Locator..... 15
- 5.12 Service ..... 15
- 5.13 Text Messaging ..... 15
- 5.14 Other ..... 15
  - 5.14.1 Export Content..... 15
  - 5.14.2 Two-Way SSL..... 15

DRAFT

# 1 OVERVIEW

## 1.1 Context

Software-based solutions can be highly complex in the capabilities and features they offer, reflecting a broad set of functions assembled from multiple discrete components. Some solutions rely on products that deliver packaged sets of capabilities (e.g., leveraging platforms that provide sufficient functionality across all feature areas with strong functionality in key areas), while others are assembled out of individual sources (e.g., built from a collection of parts, created or selected for their delivery of key features).

For each business solution, we must choose between platforms and assemblies of products, understanding the degree to which business services our customers offer are dependent/independent and similar/dissimilar. Interdependence can be assessed by evaluating which systems **act on** the same business entities (e.g., creating, modifying, deleting) vs. **act in regard to** them (e.g., approving, flagging as high risk). For example, a system used to collaboratively develop a document **acts on** the document, whereas a system used to review and approve the document **acts in regard to** it.

Comprehensive sets of processes may be decomposed into discrete segments, based on the business entities required to begin them, the business entities they produce, when completed, and the users/roles engaged in executing them. As an example, grants management proceeds through several distinct phases – recording information about an appropriation, developing a spending plan and labor forecast, creating a solicitation, collecting applications, determining which applications to award, managing progress and financial performance, undertaking audits, overseeing monitoring, and finally, closing out completed awards. While each phase supports the overall grants management lifecycle, not all act on applications or grants. Instead, they act in regard to applications (e.g., describing funding opportunities, preparing to accept applications, determining risk) and grants (e.g., determining which applications should become grants, flagging late progress and financial reports). Although some users/roles span multiple segments, each segment may be addressed distinctly, allowing solution engineers to select, develop, and/or integrate products optimized to support business processes within each segment.

In combination, segmenting business processes and identifying the points at which they act on business entities vs. acting in regard to them will enable solution engineers to define modular boundaries in the construction, maintenance, and extension of OCIO solutions.

The patterns described in this document segue with the OJP OCIO Enterprise Architecture (EA). Each application or solution identified in the EA should include information about the Capability and Feature Patterns it uses.

In the following sections, verbs like “collect”, “create”, “store”, “maintain”, and “modify” indicate **acting on** business entities while those like “consume”, “analyze”, “refer to”, “identify”, and “find” indicate **acting in regard to** business entities.

## 1.2 Purpose

This document establishes a language of standard patterns which can be used when evaluating and describing technologies and solutions and can be applied by OCIO staff and contractors in choosing and delivering solutions aligned with their business objectives.

Proposed solutions presented to the Engineering Review Board (ERB) will be mapped to patterns in this document. In cases where a business need cannot be addressed by an existing pattern or a component of a proposed solution cannot be mapped to an existing pattern but the component is identified of interest to the OCIO as an enterprise capability, ERB will oversee the creation of a new pattern, Audience

## 1.3 Audience

OCIO staff, Office technical staff, and contractors working on systems for OJP stakeholders should refer to this document when characterizing the nature of the systems they plan to deliver. When presenting

designs to the ERB, they should be prepared to speak to planned approaches to incorporate Capability, Feature, and Implementation Patterns into their solutions.

#### **1.4 Scope**

The patterns in this document address the general factoring of solutions into major functional sets to ensure that the use of technologies under OJP OCIO authority are applied in standardized and manageable ways while enhancing maintainability and extensibility.

Specific deployment architecture and software coding patterns are considered beyond the scope of this document.

DRAFT

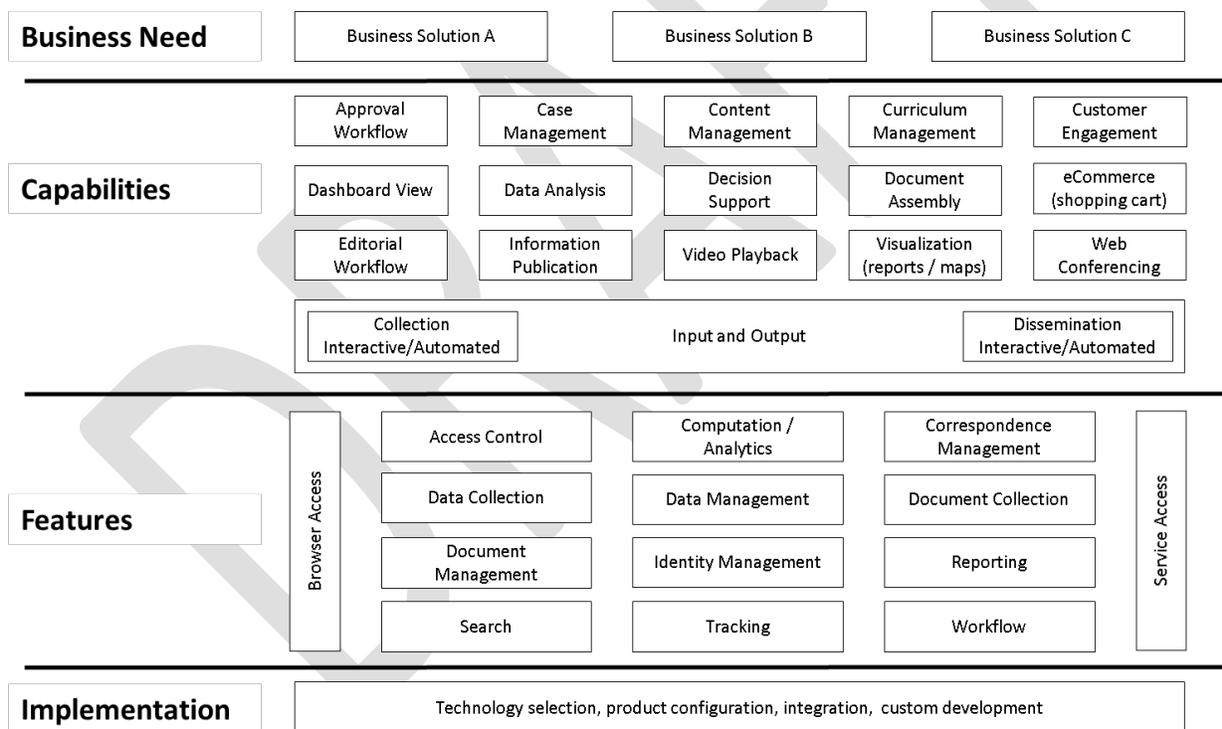
## 2 BUSINESS SOLUTIONS

Each business solution must be constructed to support and/or automate processes our customers follow to deliver services for which they are responsible. In some cases, processes are narrowly defined (e.g., risk assessment, as implemented in GAT), and in others they are very broad (e.g., award lifecycle management, as implemented in GMS and ancillary systems). Each business solution will be delivered as a coherent set of capabilities available to a defined set of users/roles.

Capability Patterns recognized by ERB are presented in the next section of this document. As an example of a narrow Business Solution that may be satisfied with a single Capability Pattern, an Executive-level Visibility solution could require only a Dashboard View. Conversely, a broad Business Solution, like a comprehensive Grants Management Solution with many sequenced lifecycle phases, must implement multiple Capability Patterns: Editorial Workflow to develop Solicitations; Decision Workflow to decide when a Solicitation is ready for release; Dissemination to promote a funding opportunity; Collection to receive Applications; Data Analysis to determine Basic Minimum Requirements; Decision Workflow (again) to support Peer Review; and on.

The following diagram depicts how Business Solutions are derived from Capability patterns, which in turn, are constructed based on Feature Patterns, themselves built through the use of Implementation Patterns.

**Figure 1 - Schematic View of Capability Patterns**



### 3 CAPABILITY PATTERNS

Capability Patterns are constructed from selected Feature Patterns to deliver coherent sets of functions assembled to address identified business-level needs.

#### 3.1 Approval Workflow

Objective: Facilitate processes leading to approval/disapproval decisions regarding and managing commentary on source information and documents (e.g., approval or denial of benefits, determination to audit an awardee, recommendation to consider an application in peer review).

Typical Feature Patterns include:

- Data Management (consume source information relevant to the decision to be made, create records capturing commentary and notes, create records documenting decisions made).
- Workflow (create and consume processes to engage stakeholders in reaching and documenting a decision or set of decisions).

#### 3.2 Case Management

Objective: Collaboratively manage a set of tasks to reach a conclusion (or set of conclusions) related to a given party or set of parties (individuals or organizations), typically achieved by applying defined business rules through a controlled process, engaging staff in assigned roles at defined stages, collecting and managing related documentation as tasks progress. Unlike Customer Relationship Management (CRM), which has some similarities to this pattern, Case Management is focused on managing a process to conclusion, while CRM is focused on managing customer engagement through the entirety of their relationship with OJP.

Typical Feature Patterns include:

- Data Collection (collect or refer to information provided by and regarding parties)
- Data Management (store or refer to collected information)
- Document Collection (collect or refer to information provided by and regarding parties)
- Document Management (store or refer to collected documents)
- Correspondence Management (create correspondence with relevant parties through multiple channels and consume contact history)

Leverages Decision Workflow (create and consume processes to pursue data, documents, and decisions).

#### 3.3 Content Management

Objective: Prepare coherent set of assets for presentation on a website or set of websites. Processes must engage users in the creation, editorial review, and approval of content. Specific information management aspects may include the ability to assign relative priority/importance of content (e.g., what information will be presented above the fold, at the top of a list, first in a carousel sequence).

Typical Feature Patterns include:

- Data Management (consume source data to merge with templates/populate dynamic pages, metadata about documents)
- Document Management (create templates and constructed web pages, and refer to PDFs, Word docs).

Leverages Editorial Workflow (create and consume processes to engage authors and editors) and Decision Workflow (create and consume processes to engage approvers in determining what content is ready for presentation).

### 3.4 Curriculum Management

Objective: Maintain coherent sets of assets available as part of a managed series. Track user interactions with assets. Control access to ensure users move from one asset to the next in a prescribed order. As appropriate, award certificates and/or credit for successful completion.

Typical Feature Patterns include:

- Data Management (create information regarding user interactions with assets and manage access in prescribed order)
- Document Management (retrieve documents in set related to a given curriculum)
- Reporting (present information showing users progress through curricula and certificates/credit for completion)

### 3.5 Customer Engagement (Stakeholder Engagement)

Objective: Understand the needs of and capture interactions with a given party (individual or organization), understand their relationship with the business, and engage them in ways that ensure a positive customer experience. Customer Engagement is similar to Case Management, except that processes continue as decisions are made about how best to engage, while processes typically conclude in Case Management, once the disposition of a case has been determined.

Typical Feature Patterns include:

- Data Management (create information regarding customer history, and consume information about their interactions with OJP)
- Correspondence Management (create correspondence with customers through multiple channels, and consume contact history)
- Computation/Analytics (create models depicting a customer's relationship and experience with OJP, helping to infer how best to serve them)

### 3.6 Dashboard View

Objective: Provide clear, brief synopses of system/business process status to management stakeholders, emphasizing KPIs and enabling management by exception.

Typical Feature Patterns include:

- Data Management (consume source information)
- Computation and Analytics (create consolidated results derived from data sources)
- Reporting (consolidate results from computation and analysis, generally indicating status relative to KPIs)

Leverages Information Publication (provide a web-based window on results).

### 3.7 Data Analysis

Objective: Use automation, analytical and visualization tools, and human insight to develop and share understandings of complex, typically large-volume data sets.

Typical Feature Patterns include:

- Data Management (consume source information, create records capturing insights, comments, and follow up questions/interests)
- Computation/Analytics (consume and analyze source data, and store patterns and trends identified through analysis)
- Reporting (consolidate and, optionally, store results of computation and analysis)

Leverages Information Publication (provide a web-based window to reports) and Service Access (provide services that client systems can invoke and retrieve analyses).

### **3.8 Decision Support**

Objective: Facilitate processes leading to decisions regarding and commentary on (typically large) sets of source data providing tools to analyze, visualize, compare, sort, filter, and otherwise manipulate data allowing users to validate insights and/or concerns about information embedded in it.

Typical Feature Patterns include:

- Data Management (consume source information relevant to the decision to be made, create records capturing commentary and notes, create records documenting decisions made).
- Computation/Analytics

### **3.9 Document Assembly**

Objective: Assemble complete documents from individually managed fragments, sometimes conditionally including sections, and replacing template fields with relevant values (e.g., creating a Solicitation out of a set of templates and interjecting data into specific fields).

Typical Feature Patterns include:

- Document Management (create and revise documents)
- Workflow (create and consume processes to collect information relevant to the construction of an assembled document)

### **3.10 eCommerce**

Objective: Allow users to find and acquire assets (e.g., publications, copies of rare documents). Present a “shopping cart” experience wherein users can select the assets they want and set up a transaction to acquire them at a cost or at no cost. If at a cost, integrate with a payment gateway to process charges.

TODO

### **3.11 Editorial Workflow**

Objective: Engage individuals or groups in defined roles participating in the creation, modification, and/or approval of a document through a staged, frequently collaborative process.

Typical Feature Patterns include:

- Data Management (create records that capture commentary in source documents)
- Document Management (create and revise documents)
- Workflow (create and consume processes to engage stakeholders in refining documents to a complete state. Note that the conclusion of Editorial Workflow may segue into Decision Workflow in the context of Information Publication to post a Solicitation)

### **3.12 Information Publication**

Objective: Offer and describe data available from our systems in ways that make clear what is available and how to access/retrieve it.

Typical Feature Patterns include:

- Data Management (consume source information created through other Capabilities, create metadata and descriptive synopses of source information)
- Document Management (consume source documents, create metadata and descriptive synopses of source documents)
- Search (locate information relevant to users' needs)

- Browser Access (enable interactive access to information)
- Service Access (enable automated access to information)

Leverages Decision Workflow (create and consume processes to engage approvers in determining what content is ready for presentation) and Dissemination, to provide access to information.

### 3.13 Video Content Management

TODO

### 3.14 Visualization

TODO

### 3.15 Web Conferencing

TODO

### 3.16 Input and Output

Input and Output capabilities act as the standard channels to collect information into OCIO solutions and to make information in our systems available to those who need it.

Input and Output patterns offer both an interactive (web browser) and automated (service) mechanism, ensuring support for human and machine-based clients, providing consistent representations of information in our systems.

#### 3.16.1 Collection

Objective: Collect data and/or documents from stakeholders relevant to business processes supported by OCIO solutions.

Typical Feature Patterns include:

- Data Collection (store information collected from users or retrieved from other systems via service calls – note that our systems should copy/replicate information as little as possible)
- Document Collection (store documents uploaded by users or retrieved from other systems via service calls – note that, as much as possible, Documents should not be copied among solutions and should, instead, remain resident and available in their systems of record)
- Browser Access (presents a web-based front end through which users can upload data and documents)
- Service Access (presents as API through which client systems can submit, and in some cases, modify or remove data and documents)

#### 3.16.2 Dissemination

Objective: Make information and documents resident in OCIO solutions available to stakeholders.

Typical Feature Patterns include:

- Data Management (consume content for presentation)
- Document Management (consume content for presentation)
- Browser Access (presents a web-based front end through which users can download data and documents)
- Service Access (presents an API through which client systems can retrieve data and documents)

## 4 FEATURE PATTERNS

Feature Patterns, individually, provide limited business-level utility, but combine in ways that deliver recognizable capabilities. Feature Patterns are derived through the use of one or more Implementation Patterns.

### 4.1 Access Control

Objective: Control access to resources based on roles associated with an identified user or client system.

Typical Implementation Patterns include:

- Authentication
- Authorization

Leverages Identity Management.

### 4.2 Browser Access

TODO

### 4.3 Computation and Analytics

Objective: Derive calculated results from source information (raw data or parsed documents).

Typical Implementation Patterns include:

- Data Storage

### 4.4 Correspondence Management

Objective: Manage messages sent to and received from known correspondents (e.g., emails, text messages) in the context of a business process (e.g., sending notice to an Awardee that Progress Reports are coming due).

Typical Implementation Patterns include:

- Email
- Text Messaging

### 4.5 Data Collection

Objective: Collect information to be stored in back end systems in the context of a business process (e.g., annotating an application, requesting a GAN, requesting a drawdown). The ability to submit some data may be limited to users and client systems that successfully authenticate and possess specific privileges.

Typical Implementation Patterns include:

- Authentication
- Authorization
- Data Storage

Leverages Browser Access (for interactively uploaded data) and Service Access (for automated submission).

### 4.6 Data Management

Objective: Store information relevant to business or operational contexts, controlling access and maintaining integrity, availability, and security. Access to some data may be limited to users and client systems that successfully authenticate and possess specific privileges. Some systems may implement automated data collection (e.g., automating requests for data from other systems, collecting log files).

Typical Implementation Patterns include:

- Authentication
- Authorization
- Data Storage

#### **4.7 Document Collection**

Objective: Collect documents to be stored in the context of a business process (e.g., submitting an application, providing documentation as part of a desk or site review). The ability to submit some documents may be limited to users and client systems that successfully authenticate and possess specific privileges.

Typical Implementation Patterns include:

- Authentication
- Authorization
- Document Storage

Leverages Browser Access (for interactively uploaded data) and Service Access (for automated submission).

#### **4.8 Document Management**

Objective: Share documents among select individuals or groups with defined roles controlling access and modification rights, potentially under version control. Access to some documents may require that users uniquely identify themselves/authenticate and possess specific privileges.

Typical Implementation Patterns include:

- Authentication
- Authorization
- Document Storage

#### **4.9 Identity Management**

Objective: Maintain individual identities and associated roles for each user with access to resources in our environment. Apply policies related to account dormancy and password ageing and complexity. Provide a single, authoritative source regarding user authentication and authorizations.

TODO

#### **4.10 Reporting**

Objective: Present selected information in requested formats (e.g., tables, graphs) that provide users with insight into data underlying systems supporting Office missions. Access to some reports may require that users uniquely identify themselves/authenticate and possess specific privileges.

Typical Implementation Patterns include:

- Authentication
- Authorization
- Data Storage

Leverages Computation/Analysis, Browser Access, and Service Access.

#### **4.11 Search**

Objective: Locate relevant documents and/or information, based on search terms specified by users or by client systems.

Typical Implementation Patterns include:

- Data Storage
- Document Storage

## 4.12 Service Access

Objective: Offer resources on the network where client systems can access it. Access to some resources may require that client systems are uniquely identified and possess specific privileges.

Typical Implementation Patterns include:

- Authentication
- Authorization
- Content Caching
- Content Distribution
- Two-Way SSL
- API-Manager

## 4.13 Search

TODO

## 4.14 Tracking

Objective: Record user and client system interactions with applications and services.

Typical Implementation Patterns include:

- Logging

## 4.15 Workflow

Objective: Define and automate progression through a series of steps/stages with clear starts and ends, engaging identified services and/or stakeholders (individuals or groups) in defined roles.

Typical Implementation Patterns include:

- Authentication
- Authorization
- Data Storage

## 5 IMPLEMENTATION PATTERNS

Implementation Patterns represent mechanisms underlying solutions and are assembled together to create Feature Patterns. Generally, they map directly to an applied technology choice.

### 5.1 API

Objective: Present services through Application Programming Interfaces (APIs) that are well defined and managed under governance controls. OCIO APIs will be defined in SWAGGER.

For business entities consumed or produced through an API, architects must represent which must adhere to DATA Act standards

Enables the Service Access pattern.

Leverages API-Gateway.

### 5.2 API-Gateway

Objective: Provide a layer of abstraction between clients and services, so that access to services can be secured, managed, and measured centrally.

Manages use of APIs.

Leverages Service Access.

### 5.3 Authentication

Objective: Enable global assurance among OCIO systems that users are who they assert themselves to be, and establish trusts between clients and services.

### 5.4 Authorization

Objective: Determine whether an authenticated user or client system may access requested resources.

### 5.5 Content Caching

Objective: Decrease resource request response time within OCIO network environments by retrieving information from source systems only if the requested information had not been retrieved, previously, or if it has changed, since the last request.

### 5.6 Content Distribution

Objective: Decrease resource request response time outside of OCIO network environments by caching resources at “edge nodes” near users or client system that issue requests.

### 5.7 Data Storage

Objective: Maintain data supporting business processes and system operation in a secure, stable, consistently available state.

### 5.8 Document Storage

Objective: Maintain documents supporting business processes in a secure, stable, consistently available state.

### 5.9 Email

Objective: Send and receive email messages between OCIO systems and known correspondents related to business contexts and processes.

## 5.10 Logging

Objective: Capture automated activities and user actions within OCIO systems in a form that can be collected, interpreted, and understood by automated tools.

## 5.11 Resource Locator

Objective: Provide a stable, logical reference where resources can be accessed and/or submitted. The pattern for Resource Locators are structured so that OJP OCIO can maintain awareness of and security control over all resources for which it is responsible.

Pattern: **<product>.<office>.ojp.gov/<resource\_path>** where <product> represents a deployed website or other resource root (e.g., facjj), <office> represents the Office responsible for the website or resource (e.g., ojdp), and the resource\_path represents a particular resource or resource type within a site (e.g., "awardees", "index.html").

Note that, for non-Program offices (e.g., OCIO, OCFO), the <office> element will not be used.

Examples:

- facjj.ojdp.gov
- gms.ojp.gov/awardees

## 5.12 Service

Objective: Deliver value from OCIO systems by providing access to the resources they represent (e.g., information, calculations, documents) through a consistent, shared mechanism.

Leverages API as a standard representation of available resources.

## 5.13 Text Messaging

Objective: Send and receive text messages between OCIO systems and known correspondents related to identified business processes and contexts.

## 5.14 Other

Patterns in this section are either dependent on policy, and will be promoted to have a primary place among Implementation Patterns, once the policy has been established, or are intended as temporary measures, and will be replaced once permanent measures are in place.

### 5.14.1 Export Content

Objective: Preserve bandwidth on the JMD/OCIO network by moving large content (e.g., images, video, audio) to external SaaS providers.

### 5.14.2 Two-Way SSL

Objective: Support mutual authentication between clients and services. Disallow connections if mutual authentication cannot be established. Note that this is intended as an interim pattern, until the OJP OCIO API-Manager is in place, at which point access to services will be allowed only to clients with allocated API-keys.