Application Reference Architecture

Table of Contents

Document Generation Date: 2022-04-24 10:04

Introduction

This document outlines the Application Reference Architecture (ARA) as it applies to our department.

What is architecture in general?

- *Architecture is the stuff you can't Google.* - Mark Richards.
- *Architecture is the decisions that you wish you could get right early in a project, product or project lifecycle* - Ralph Johnson & Martin Fowler
- *Architecture is about the important stuff, whatever that is.* - Ralph Johnson & Martin Fowler

The Application Reference Architecture (ARA) borders on what many would consider an enterprise reference architecture. This document, the ARA, attempts to provide an overview of the enterprise environment with a focus on application architecture elements. - Enterprise architecture documents the whole architecture and all important elements of the respective organization, covering relevant domains such as business, digital, physical, or organizational; and ii) the relations and interactions between elements that belong to those domains, such as processes, functions, applications, events, data, or technologies." - [Wikipedia - Enterprise Architect].
- Application architecture describes the behaviour of applications used in a business, focused on how they interact with each other and with users. It is focused on the data consumed and produced by applications rather than their internal structure. In application portfolio management, applications are mapped to business functions and processes as well as costs, functional quality and technical quality in order to assess the value provided." - [Wikipedia - Application Architect].

This document documents: - existing application architecture within our department - guidelines for technical leaders

This document is intended for: - technical design leads

Out-of-Scope

- This document is neither a vision, nor a strategy nor a roadmap document.

- This document is neither a strategy, nor a department culture nor an development process document.
  - Strategy: What will and will not do, and how govern resources.
  - Culture: People, Processes (Organization / Teams), Communication
  - Execution: Processes, Tools

Business

A common way for the business to communicate what the organization needs and does is through a business capability model (BCM). There are many uses for a BCM. Product owners can use a BCM to drive convergence in technology and business processes to enterprise standards. Regular review of aligning the BCM with the department strategy and vision can allow enterprise architects and business architects to identify and prioritize the corresponding IT initiatives with business needs. Internal committes, working groups and forums can collaborate to identify reusable business process and push for adoption across the organization. Business capabilities, processes, information flows and value streams should be assessed routinely based on efficiency, priority, and complexity.

Our department has a Business Capabiltiy Map (BCM) describing the main capabiliies required to fulfill our mandate. To help support the business our technology teams provide a broad range of IT capabilities. Our IT department supports many networks both nationally and internationally. Within the IT department, our software development team supports an extensive catalog of applications.

The health of our portfolio needs to improve as identified in our Corporate Risk Profile (CRP). Several leadership principles have been established over the years to provide guidance when addressing business needs. Key principles relating to directing architecture and design are:

1. Rationalization: We have an long queue of valuable business requests and opportunities. During the software development phase, requirements must be rationalized against the original approved project scope and other compete busines needs. The costs of increment application development, both in project costs and ongoing costs must be carefully understood. This is the process of rationalizing business needs and can include the senior management team when necessary. [See Guidance - Rationalization for more informatoin - !!!]
2. Executive Lead / Change Management: Projects and programs need executive sponsors who are committed to the change management and ratinalization required to allow IT to develop a product.
3. Business Architecture and Artefacts: The business plays a key role in shaping the application. Business architecture (capabilties, value streams, information flows, organization model) are essential for successful analysis of the business needs during application development. Significant architecture re-work and design waste result if these are unavailable.

The following are useful: - Business Capability Model (BCM) - [Wikipedia - Business Capability Model] : A diagram that identifies the business capabilities with regards to the application being developed. The GC BCM is a reference, and our department has an internal BCM. The BCM traditional is decomposed into 3-4 levels with descriptions of each level. The application requirements are mapped to the respective BCM capabilities. - Value Streams - [Wikipedia - Value Streams] : Introduced in Lean (1950's) a value stream is a set of actions (workflow) to produce value. Value Stream Mapping [Wikipedia - Value Stream Mapping] is visual tool introduced in Lean Management methodology to display the value stream with define icons to show delays and inventory stages. An example value stream might be recruitment "street to seat" or "hire to retire". - Information Flows [Wikipedia - Information Flow Diagram] is a business view of how information flows between business responsiblity centres. *The main purpose of an information flow diagram is so that sources that send and receive* *information can be displayed neatly and analysed.* - Organization Model / Operating Model [Wikipedia - Operating Model]- An operating model is both an abstract and visual representation (model) of how an organization delivers value to its customers or beneficiaries as well as how an organization actually runs itself. The elements that make up the Operating Model are People, Process and Technology (PPT).

1. Business Requirements: Business requirements for IT analysis, prioritization and design. Business requirements should attempt to be specific, measurable, achievable, realistic/relevant and time-based (SMART). By being SMART, this affords the technology project the ability to effectively plan and analyse.

As our department adapts agile methodologies, incremental value in the project can be obtained by the agile team developing business artefacts.

Technology Environment

Our IT operates in a complex constrained environment.
!!! - add on corporate production details !!!

Discovery

There are many initiatives within our department that require enterprise and domain architecture effort to recommend the path forward.

1. Identity and Access Management (IdAM):
   Analyze existing identity and access management options to provide multi-domain identity and access to compartmentalized information.

2. Enterprise Integration & Interoperability:
   Analyze steps to mature our ability create a composable enterprise Gartner with an focus on leveraging modern API concepts (API management, API catalog, API developer experience - sandbox, versioning, . . .).

3. Enterprise Search:
   Gartner calls the broader enterprise search an Insight Engine. Gartner - Critical Capabilities for an Insight Engine. [Gartner Magic Quadrant - Insight Engines]. Key terms include; connectors, touch points, integrations. Popular open-source solutions like Solr and Elastic support API integrations for adding and removing content with structured-metadata. A key to the success of enterprise search is the ability to structure the index information with metadata. This enables discover and faceted searches.

- [ ] Enterprise Taxonomy : A deliverable within the Information Management Modernization program (IMmod)

1. Multi-Security Zone Applications:
   Our directorate has been asked to to move workloads to lower security

zones. As a consequence business processes may span security zones. The cross-domain-solution has been identified as an enabler technology. What overall application, data, information and security architecture is needed to realize these benefits.

2. Managing Media Our department manages multimedia (images, audio and video files) as well as file-types on a diverse range of applications. Media management can be addressed be a Media Management Platform and a Digital Experience Platform (DXP). Industry leaders include OpenText, Oracle and Salesforce. While some Content Management Systems (CMS) also support DXP features many new market entrants are SaaS-based and require cloud connectivity (e.g, Sanity.io). OpenText DXP() is not in the top-magic quadrant; however it deserves consideration due to GCdocs.

- OpenText DXP
- Opentext Why you Need a DXP: *Orchestrating a cohesive, contextual experience that meets brand standards, achieves business goals across all channels and touchpoints, while it delights the recipient, is a massively difficult task.*

Features of a DXP: - Content Management System - Media Asset Management - Digital Asset Management (media and non-media content) - Headless DXP / CMS : Provide back-end features expose media assets via API's.

TODO - reference Confluence ITOD Dependencies document TODO - Add Enterprise Interoperability to ITOD Dependencies

Application Characteristics and Styles

- *Architecture Style: The combination of distinctive features related to the specific context within which architecture is performed or expressed; a collection of principles and characteristics that steer or constrain how an architecture is formed.* - TOGAF

Application Characteristics

As part of the analysis and design some high-level characteristics of the application should be assessed. Some of these attributes may be official documented as part of the project and application development, and others may have to be assumed or derived for requirements

Attribute	Description	Note
Criticality	How critical is this application to the business. This is sometimes referred to as Tier-1, 2, 3.	The department lacks an official list of application criticality. Based on criticality, and TBS guidance, critical applications must have certain quality components like a business continuity plan (BCP) and a Disaster Recover Plan (DRP). This need to maintained and practised like fire alarms on a regular basis.
Security Profile	Based on the security triad of Confidentiality, Integrity and Availability (CIA) and indicating the impacts of integrity and availability to the organization (High, Medium, Low).	Common profiles are PBMM (Protected-B, Medium, Medium) and TSHH (Top Secret, High, High). The security profile can help guide development of quality requirements (non-functional requirements)
Information Classification	What classification of information is managed by the system	Unclassified, Confidential, Protected A/B/C, Secret and Top Secret are common security classifications
IM Repository Type	Identifies whether the information in this system is transitory or corporate.	Based on the repository type additional requirements relating to managing the information through its lifecycle are required. Reference Guideline on Service and Digital

Attribute	Description	Note
Information Business Type	Our department treats operational information different from administrative information.	The distinction is unclear, and there are few guidelines to help projects to help manage this distinction. Applications are categorized as managing operational or administrative information. For example, CW is administrative, CWOPS is operational (however only extremely limited operational information is permitted in CWOPS).

Application Architecture Styles

Architectural style is defined as a set of characteristics and features that make a building or other structure notable or historically identifiable. Architecture styles are been established and evolved over the years. Some common application architecture styles are [Fundamentals of software architecture]: - [Richards, Mark. & Ford, Neil. Fundamentals of software architecture: an engineering approach. (O'Reilly, 2020)]: - Distributed: Microservices Architecture : pros (reliablity, modularity, elasticity, +++), cons: (cost, complexity, ...) - Distributed: Orchestration - Service Oriented Architecture (~2005) : pros (good elasticity, fault tolerance, scalability), cons: (complexity, testability, cost, ...). A big weakness of SOA was the use of a common platform for all services deployed (e.g., Oracle SOA Suite, IBM WebSphere, DataPower, MessageBroker). SOA also required stateful services and sharing of context (tight-coupling). Note, SOA promised loose-coupling, scalability and fault tolerance Josuttis, N. M. SOA in practice. (O'Reilly, 2007)] however these were difficult to achieve. - Distributed: Event Driven Architecture : pros (fault tolerant, modular, good cost), cons: (complexity, testability,) - Monolithic: Layered: 3-tier/N-Tier/Client-Server : pros (simplicity and cost), cons: (scalability, fault tolerance, deployability, testability, modularity) - Monolithic: Pipeline: pipelines & filters : pros (simplicity and cost), cons: (scalability, performance, ...)

TODO Discuss architecture styles and key characteristics - ../Downloads/The SOA Journey_ from Understanding Business to Agile Architecture, and how this leads us to loosely coupled, high-cohesion and encapsulated architectures (EDA & Microservices).

Application Architecture Guidance

Goal: Composable Enterprise

1. Integrations / Messaging Methodology: DDD / Bounded Context !!! : what was the API guideline - do one thing - either a command or a query but not both… - e.g., an API should get() info but or set() info but not both…???where was this.

Goal: Reduce Technical Debt

1. Rationalization.

- … - Feature Selection - Start with No
  - reduce features, focus on the priorities
- Steve Jobs - Focus on Saying No - 1997
  - *Apple suffered from lousy engineering management* - Steve Jobs answering a negative question about a removed feature. Focusing is about saying no. *When you say no, you piss off people.*
- Basecamp - Priorities - Whats the Big Idea
  - Are we staying true to the vision?
- Basecamp - The Starting Line - Fix Time and Budget, Flex on Scope
  - Prioritize, Focus on what you really want to deliver), Flexibility : Scope flexibility. It's better to make half a product than a half-assed product (more on this later). *How does a project get to be a year behind schedule? One day at a time.* - Fred Brooks 1979 Software Project Management - The Mythical man Month. the mythical man-month - 1975 - isbn

1. Reuse / Buy / Build.

- Reuse: Attempt to reuse what we currently own, or what other government departments / partners are using.

- Buy: Buy solutions and integrate into our enterprise architecture
- Build: As a last resort, custom build a solution. This should be limited to business capabilities and processes that are unique to our department. Executive approval (Department Architecture Review Board) required.

1. Document & Exercise Backup & Recovery All applications, regardless or criticality, must have a documented backup and recovery procedure. This needs to be exercised on a regular basis (at least annually) and must be done prior to deployment to production.

Business critical applications require a BCP and DR plan to be documented and reviewed on a regular basis. - [ ] Enterprise Architecture : Formally identify the criticality of applications and record this in the deparment's official configuration management database (CMDB).

Goal: Reduce Content Duplication.

Content is duplicated within applications and across technologies. The causes of this have not been formally documented, however some factors leading to users copying content are the lack of *trust* in being able to find or access the content in the future. This can be paraphrased as *I need a local copy for me or my team.* This leads to copies of information on shared-drives and transitory and corporate applications. Some historical examples that have led to this "clone-and-own" culture include: - Link Rot: Application upgrades making links to content fail. Deep Linking is the use of a hyperlink that links to a specific, generally searchable or indexed, piece of web content on a website. For example, a link to a specific case, request or document.
- Access: Users are concerned that the content may disappear due to the content owner removing, renaming or modifying user-access. This is difficult to address at the application layer, and requires enterprise information and access-management governance.

1. URL Lifecycle When supporting Deep Linking design must take into account the lifecycle of the link, and its ability to function through upgrades. Consider patterns such as Permalink and Data Object Identifier (DOI). When provide a link to a user for reference, identify this should be a trusted-link which survives upgrades/replacements.
2. URL Design Define a URL strategy for the application, including an inventory or URL's provided. Define the manner in which URLs are clean, friendly and pretty Clean URL; support *http://example.com/name* as opposed to *http://example.com/index.php?page=name.*
3. Enterprise Search Enterprise search will definitely help in enabling users to find the information they should have access to. This is a major long-term initiative.

References

TODO 37signals - use as support for guidelines *TOD* CIO - use as support for guidelines

Basecamp - The Starting - Build Less [. . .] less means: - Less features - Less options/preferences - Less people and corporate structure - Less meetings and abstractions - Less promises

[. . .] Basecamp - Stay Lean - Less Mass: - less "Thick process" - less "Long-Term Roadmaps" (supported as by ITSS Study - Ian Lovsion 2017) - less of "The past ruling the future"

Goal: Composable Applications

Applications should reduce coupling; especially at the high-level interactions between components. Reference: Reduce Coupling - Martin Fowler IEEE 2002.

Architectural patterns to support composable applications include: *TODO* are these patterns or concepts?? 1. High Cohesion ref oreilly - chapt 3 - modularity 1. Low Coupling : why persistence layers / data access layers added between application business logic and the database layer.

Principles supporting composable applications: - SOLID : S - Single Responsibility Principle (Martin, J. Principles of object-oriented analysis and design. (Prentice-Hall, 1993)

```
If a class has more than one responsibility , then the responsibilities become co
Changes to one responsibility may impair or inhibit the class ' ability to meet t
This kind of coupling leads to fragile designs that break in unexpected ways whe
```

1. Decoupled Integration
2. Develop an API Strategy API's are a critical component of our technology stack. As applications and technology more-and-more through API's we need to mature our API strategy. The API Strategy should address concerns such as:

- API Discovery / Catalog: How can developers discover integrations (*TODO*)
- API Testing: automated testing, performance testing, stubbed-out testing.
- API Standards follow GC Standards on API guidance, align with NZ API Guidance & Resources & UK API Technical & Data Standards guidance. These are written to support integrated digital processes across departments and agencies; however their guidance is relevant for internal integrations.
- As we mature with our API Strategy, and enterprise approach to APIs for the following is important:
  - API Documentation(https://www.gov.uk/guidance/how-to-document-apis): discover, affordances (understand how to use API), integration with API. Examples: GOV.UK Frontend, Stripe API, Mailchimp.
  - API Protocols: Leverage protocols and languages like gRPC and GraphQL for integrations.
  - API Management: As the number of components, micro-services and integrations grow, the need for an API management layer to provide orchestration and API lifecycle management increases. API management provides a single point of entry for all connected systems and services. Helps developers (IT, client-authentication, authentication, business-citizen) develop to APIs.
- References:
  - Wikipedia API Management
  - Gartner Ensure API Management includes Cloud and Microservices
  - Gartner Human Capital Management (HCM) Integration Strategy - 2020
  - Gartner Choose Integration Technology - 3 Patterns of Integration: 1) Data Consistency across platforms (ERP, CRM, Billing, . . .), 2)

Multistep Process / Pipeline and 3) Composite Service. Recommends to identify integration needs (Application Integration, Data Integration, API Lifecycle Management, Integration Platform, BPM (Pega, ..), Master Data Management, Message Oriented Middleware (ESB, Streaming, . . .), Robotic Process Automation(RPA))

1. Event-Driven Process and Streaming Leverage events as a core principle. Publish these events, subscribe to these events (streaming data flows).

- event-based data flows for batch and real-time prcoessing
- messaging-oriented over transactions. Interested Video Presentation on Why - Avdi Grimm - Nordic JS No Return: Moving beyond transactions

References

- Gartner - The Future of ERP is Composable : Composable ERP is defined as an adaptive technology strategy that enables the foundational administrative and operational digital capabilities for an enterprise to keep up with the pace of business change. This strategy delivers a core of composable applications and, as a service, software platforms that are highly configurable, interoperable, and flexible to adapt to future modern technology.

  Recommendations:

  - Defining business capabilities through road mapping exercises
  - Reduce reliance on customizations and proprietary-ERP
  - Continuously deliver incremental business value in a modular mode
  - Build an ERP team with wide-ranging skills (~DevOps for ERP); skill development and training

Goal: Testable Applications

Testing applications and groups of applications effectively and efficiently requires anlaysis and design. The application development needs to include capabilities to facilitate testability. Testing scopes vary based on developer testing and quality assurance testing. QA testing often involves elaborate efforts to setup a system (install, configure and provision) for a single test case.

- Automation: Adopt the DevOps of shift-left the integration and quality testing. The automation should accommodate the CI/CD concepts, as well as the ability to provision and validate tests.

- The following principles are copied from Microsoft [Shift Left Testing].

  - Write tests at lowest level possible. Favour unit tests over functional tests. When functional tests fail, consider if unit tests should be more comprehensive.
  - Write-once, run anywhere (DRY - Do Not Repeat yourself): Tests should be written to work in any environment (Dev, Sig, Prod).

  - Design Product for Testability. Discuss how the system is testable during peer-reviews and Technology Review Board (TRB) reviews.
  - Test Code is a product. Treat the software used to automate testcases as code. The code is version-controlled, and discoverable (i.e., it exists in close proximity to the application code
  - Test ownership follows application software ownership. The software development team owns creating automated tests for not only unit-tests but boundary/integration tests.

- [ ] SDLC Checklist: FY 22/23: The Quality Assurance Working Group and the SDLC Working Group should consider formalizing the above principles and guidance as part of the new SDLC process, milestones and checklist.

Development methodologies like test driven development (TDD) that predates DevOps. How they are related is well described in this article - [TDD for a DevOps World]; summary: - TDD is clearly a quality enhancing practice. It's a really good way to mitigate the risks of defects and to increase the chances of actually the sort of rugged code that needs to withstand change and increasing demands of a DevOps world where expectations are much higher.

Goal: Future Proof Technology

The MACH Aliance was announced in December 2021. AWS, MongoDB and others are associated with this alliance.

This manifesto is:

```
"Future proof enterprise technology and propel current and future digital experi
```

The GC CIO provides guidance, and these should be respected. MACH aligns with the GC directions. - **M**: Micro-Services (Modular): Individual pieces of business functionality that are independently developed, deployed, and managed. A swappable architecture. - **A**: API: All functionality is exposed through an API. - **C**: Cloud: SaaS that leverages the cloud, beyond storage and hosting, including elastic scaling and automatically updating. - **H**: Headless: Front-end presentation is decoupled from back-end logic and channel, programming language, and is framework agnostic.

Our constraints may limit our ability to leverage the cloud. For on-premise constrained systems DevOps practices to automate updating and Kubernetes autoscaling should be prioritized.

A view of how MACH applies to guidance and industry patterns is below.

MACH	Description	ARA Guidance	Industry Paterns
M - Microservices	Individual *pieces* of business functionality can be deployed and manageed	Composable Applications	Microservices Architecture Style
A - API	Functionality exposed via API		GC API-First, GC Standards on API
C - Cloud	Leverage SaaS to its fullest including scalability and automation		
H - Headless	Decouple front-end from back-end		

Aside: An interesting article on how a CMS is attempting to become MACH-compliant; with discussion on impacts to CMS features like editors, and the use of technologies like JAMstack. MACH Sitecore Architecture

Goal: User Experience

Our user experience can be improved by looking at modern applications and their integration into varying computing platforms (desktop, mobile, tablet). Some modern experiences can include: - Push Notifications: Business fit-for-purpose notifications using Push API and integrated into Windows Operating System experience. Replace mindset of email-based notifications into a notification platform with end-user ability to control notifications. - Sharing Content Across Platforms: Ability to share content across platforms similar to sharing news and social-media content. Allows the ability to communicate effectively in different channels (intranet, CMS, . . .). oEmbed

Goal: Accessibility

The Accessible Canada Act received Royal Assent on June 21, 2019, and came into force on July 11, 2019.[Reference]. Our department has no formal policies on accessibility. In light of no policy, applications should strive to achieve WCAG 2.1 Level AA. This goal changes by application, and development must ensure they are aware of the business requirements for accessibility.

WCAG 2.1 Level AA (Double-A) implies: - Media: Captions are present on live video. When appropriate, there exists audio description of what's happening on streaming media. - Markup: Ability to resize text without breaking layout.

Language is declared in document. - Design: A minimum contrast of 4.5:1 among elements. Heading tags (h1,h2,h3, etc.) are present and emerge from content organically. - Forms: If an error is present on a form, the website will: suggest ways to fix it, the user may withdraw and resubmit the form, or the form prompts a confirmation. - Navigation: Pages can't be nested or unintentionally obfuscated unless part of a step-by-step process, such as an application or feed result. Navigation follows a semantic structure and is repeated on pages.

Applications should allow users to self-identity accessibility needs.

TODO Add content from high-side Confluence at work.

Other - TODO

Goal: Streaming-Based Application

TODO Streaming-Ready/capable application - https://www.dbta.com/Editorial/Think-About-It/Building-a-Modern-Data-Architecture-for-the-2020s-148239.aspx?PageNum=3

Goal: GC Alignment

1. CTO - Government of Canada Digital Standards : Design with Users, Iterate and improve frequently, Work in the open by default, Use open standards and solutions, Address security and privacy risks, Build in accessibility from the start, Empower staff to deliver better services, Be good data stewards, Design ethical services, Collaborate widely
2. Cloud Adoption

Goal: Architecture Strategy

Creating a Strategy

The architectural strategy for a program; whether they are renewal efforts (ERP, HR, IM, Collaboration) or greenfield (Case Management) should follow methodologies proven to be successful.

The guidance below is a summary of **Technology Strategy Patterns**. [Hewitt, E. Technology strategy patterns: architecture as strategy. (O'Reilly, 2018)]

Apply Patterns to Formulate a Strategy

- Context: Trends, Constraints, Stakeholders
- Understand: Research, analyse and understand your stakeholders, the environment and the technology landscape.
- Options: Identify options in the products, services and technology roadmaps
- Analysis: Analyse options.

- Recommendation: Make recommendation and obtain approval.

Concerns of an Architect

- Contain entropy: Show a path in a roadmap; garnering support for that vision through communication of guidelines and standards; and creating clarity to ensure efficiency of execution and that you're doing the right things and doing things right.
- Specify Non-Functional Requirements / Quality Requirements: The "..ility" list. scalability, availability, maintainability, manageability, monitorability, extensibility, interoperability, portability, security, accessibility, observability, conformity (laws, directives). Wikipedia - Quality Attributes
- Determine trade-offs: Identity the least-bad option.

1. Patterns / Tools The book goes in significant theory like propositional theory; and the advice is aimed more at the private and political sectors. The intent is to create an hypothesis and then validate it formally.

Some tools suggested are: - Logic Tree - SWOT - Strengths, Weaknesses, Opportunities and Threats - Ansoff Growth Matrix - Harvard Business Review 1957 : 2X2 matrix with Market and Products with values of Current/New.

Ansoff Growth Matrix	Market-Current	Market-New
Product	**New**	**Current**
New	Market Development Strategy: Develop new markets for new products	Diversification Strategy: Develop new products in new markets
Current	Market Penetration Strategy: Gain market share with current products and market	Product Development Strategy: Develop new products in current markets.

Corporate (Enterprise) Context

Position the enterprise for competitive advantage. - Stakeholder Alignment : A top-down approach, determine the organization chart, and *determine what leaders at the VP, Senior Director, and/or Director level matter in terms of your strategy.* Based on the stakeholder matrix determine their influence and impact and associate an approach to working with them (monitor, keep informed, maintain confidence, collaborate). - RACI - Life-Cycle Stage - Value Chain : Identify where value is created (hint: legal, intrastructure, IT, HR, procurement are supporting the value chain.). Maximize efficiency and value. - Growth-Share Matrix - Core/Innovation Wave - Investment Map

Patterns

Patterns are known, proven solutions. Patterns help us communicate architecture and design to each other.

Drive strategy with patterns. - application architecture patterns - software design patterns : Decorator, Factory, Visitor, Pub/Sub, .. - user experience patterns

Architecture Styles

Big Ball of Mud - Anti-Pattern: An application without structure, software making direct database calls, with no concerns for deisgn. In 1997, Brian Foote and Joseph Yoder, coined this the Big Ball of Mud:

```
A Big Ball of Mud is a haphazardly structured, sprawling, sloppy, duct-tape-and-

The overall structure of the system may never have been well defined.
-Based Application
*TODO* Streaming-Ready/capable application
- https://www.dbta.com/Editorial/Think-About-It/Building-a-Modern-Data-Architect
If it was, it may have eroded beyond recognition. Programmers with a shred of ar
```

User Experience Patterns

Many patterns exist for a successful user-experience (search, navigation, filters, comparisons, grids, ...) - CodePros - Patterns - Google Material UI

References

References

DJN Test

Software

Architecture

- Richards, Mark. & Ford, Neil. Fundamentals of software architecture: an engineering approach. (O'Reilly, 2020)

Design

- [1.Vernon, V. Implementing domain-driven design. (Addison-Wesley, 2013)] http://www.worldcat.org/isbn/9780133039900

Patterns

- Hewitt, E. Technology strategy patterns: architecture as strategy. (O'Reilly, 2018)
- Design patterns: elements of reusable object-oriented software. (Addison-Wesley, 1995).
- Hewitt, E. Technology strategy patterns: architecture as strategy. (O'Reilly, 2018).. Analysis, Strategy Creation and Communication Patterns. Audience is technical leads and architects attempting to recommend a strategy.

Principles

- Martin, J. Principles of object-oriented analysis and design. (Prentice-Hall, 1993)

Government of Canada

- CTO - Government of Canada Digital Standards : Design with Users, Iterate and improve frequently, Work in the open by default, Use open standards and solutions, Address security and privacy risks, Build in accessibility from the start, Empower staff to deliver better services, Be good data stewrds, Design ethical services, Collaborate widely
- GC Information Management Guidelines - 1996
- GC TBS Information Management Strategic Plan - 2017-2021: Includes strategic goals and objectives.

Terms

Value Streams: See also Scaled-Agile Framework - SAFe - Value Streams. Agile Release Trains (ART) within each value stream develop the business solutions used by the operational value streams. ARTs are cross-functional and have all the capabilities—software, hardware, firmware, and other—needed to define, implement, test, deploy, release, and where applicable, operate solutions.

TODO

TODO 2022-04

Themes : API, consistency, cohesive

Architectural Examples : To demonstrate architecture, c4model, ... (mach model) 1. Collaboration: What is collaboration? What is it not? - Contribute and read centrally. e.g., VISIOPS - essentially List management but done centrally with CW - Collaboration is not "the intranet"

1. Strategy: ....

2. Collaboration Capabilities

- share, version control, notify,

1. Technology supporting strategy:

- Departmental : CW, Confluence, GCdocs, Drupal, ...
- IC / 5EE : BallistaPlus
- task management : ITBM-PM, JIra, ...
- workflow :

1. Key Pillars to Success

- Governance: Major IM concerns about sprawl of transitory information.
  - Sprawl, Lifecycle, Admin vs Operational workoads
  - IM, eDiscovery, Hold, Disposition
  - ECM integration, enterprise taxonomy
  - search, access,
  - L&D
  - IS audit, access control, anonymous access,
  - operations: monitoring / NOC

1. Legacy ITSS Pillars (2012) JP Lachance

- Quality, Communications, Training, Operations

Roles - full stack developer - back end developer - systems integrator

Agile - SAFe - LEAN : WIP Limits, Visual Display, Feedback Loop - Scrum : Retrospective, Backlog, Grooming, Poker (called??)

API's - governance : between domain-models. HR, ERP. Smaller domain-Models - master data (shared data across many systems), system of truth

Ux - GCcollab

Navita - work backwards - future proof

Decision Making - Problem Solving, Kepner-Trege. The decision might not be right, but did I go about it in a skilful way.

React - Why React .

Architecture - 4 Parts (not sure if from SAA-C02 Pearson AWS Certification test - on O'Reilly) 1. Architecture versus Design - How to collaborate with development teams to make architecture work.

2. Wide Breadth of Technical Knowledge with some Technical Depth

- Allows architect to see solution, problems and options that others do not server

3. Understanding, Analyzing and Reconciling Trade-Offs Between Solution & Technologies

4. Understanding Importance of “Business Drivers” and how they translate to architectural concerns.

Culture

- Westrum Culture: Useful, Timely, Relevant

BI

Tableau

- Anaylitcs, Calculations, Data Prepparation, Data Relationships, Data Visualizations, Excel and Tableau, Performance, Storytelling, Tableau Server.

DAaaS

- Kubeflow

Data

- DRA : Pragmatic, Feasible, “I can build it, consistently.”. Clear guidance to get of debt. How to get out of debt, moving forward.
  - for initiatives have you gone to forums (TRB, AWG, . . .)
  - other bodies: GC, EDO, CIO, CSO, EPMO, FIMB, SMB, PM, IMB, CMC
  - pillars to move forward
- mutable data

Lower Priority

DevOps - Puppet vs Ansible vs Terraform - Travis, Jenkins, . . .

Data Mesh

ML Pipeline Architecture. Policy on Automated Decision Making.

Notes - Paper to Write-Up

- GraphQL / RESTful
- microservices,
- REST HATEOAS
- Scrum, LEAN, LEAN Six Sigma
- OReilly SW Arch Patterns - monolith, etc
- Raman - Arch Phone-Call 2022-01-12

- Had talk with Mike and he likes API approach and wants to be included
- Had talk with Amy and Raman was unable to communicate my ideas of API gateway and the value
- Amy mentioned CMS / DL review with TRB and pattern exists to expose things into lkae.
- Amy we have a pattern for cms to lake that altor should follow. If this pattern does not align with my Vision for api management… …timely for better approach..chat with Amy
- Ent search ODP
- Ent Search of everything….pattern…Ent layer…Build special index……..LRS Extract…
- Leverage Gartner.
- Enterprise pattern push i n discovery
  - …by SPM
  - …what is MAM…open-siurce MAM, image Strategy
  - * …by Archs…tech media management, search, …
  - …install on-prem…
- Reference Arch
  - look and know how to integrate to lake, to ecm how expose data to catalog, BI,. Search
  - TRB to Lake via REST API…
  - …Q'S does it reflect API vision…
  - Amy meaning of API management…talk to her… focus of API management
- What is Important arch to invest in…
  - IdAM…
  - Palantir access.
  - CMS model different classifications can see 90% of case….HS case….
- EDA
- Help projects to skeleton to build projects to.
- Side of desk….discovery roadmap to focus….
- ECE
- Raman - Arch Bi-Lat 2021-12-16

1. HR mod
2. Ref Arch - mature organizations

3. Bricks
4. PB U/L
5. CMS / P-app
6. CMS - Drag & Drop
7. CBSA SSA
8. Partnership - Corey, Jackey Vanessa - SEL MERITAGE, OPSCOM

- Guidebook Process, Principles
- IM, Access, IdM, ABAC, s.15 access control
- ICIS model / keystone model

- Awareness of past, problmes, guide, problems it created, what to look out for
- adhere to principles
- ideas - Navita - c4model
- multi-security zone
- Amy said talk to Francis - Vision, Containers, AppDev, CI/CD
- Apps/Data - Arteryx
- Access Control - Hashicorp / Vault

1. CMS not deploying in March. Vanessa said not ready. Release Mangemetn Plan - not time driven, want more features.
2. UK Digital Services & Technoloyg (DST) Strategic Plan - 2018 - ICT Strategy 2018 - IaaC 23124
3. Google Zanzibar. Lighthouse - measures site for accessibility.
4. OR - 3335 ORS by 1 person in 6 monhts. Client of Intelligence - CAF
5. Task Management : What is solution? Responded, who are the tasks being managed by (i.e., is the user-base within department or OGD - does solution need to cross air-gaps)
6. Raman: 4 Pillars:

- Consistencys: leverage same architecture, design patterns to help DEV drive consistency & future staff mentoring .
- Integrit, Robustness, Consistency
- Observability
- Data Driven Testability

1. Air Gap, DCI, On-Premise, Access Model (Case File, CodeName), eCDS, tokenization (Rube Goldberg idea of Workday)

- CMS, DL, P-App : LRS extract, visibility labels, system of record for entity
- Ref Arch : c4model
- Patterns: 5EE, EA, Fowler, … , 12factores (web-app) SOLID, Design Pasterns - Booch, OWASP, CCCNS (CSE Cyber Centre)
- Patterns; BFF / FE/BE, ..
- What affordances has our culture provided you?

- least wrong answer, most right answer : justification can be "developer experience"

- Guidance: Courts, Classified Information, . . .

1. Alpinist : Marc-Andre Leclerc
2. PPT - People, Process, Technology
3. Data-Sentinel

- Upsolver, Snowflake (Kafka, AWS Kenesis,) -
- Databricks (. . . /drive/. . .)
- ../Downloads
- Data Sentinel - GRC / GPDR /
- Gartner - How_to_Document_Appl_750245_ndx.pdf (../Downloads)
- Good-Cheap-Fast.jpg (../Downloads)
- 2017 - Martin Kleppmann Designing Data-Intensive Applications(2017)
- Dataiku-DataLabelling.pdf

1. NIH (Raman shared 2018 strategy) - more surfing:

- NIH - EA Management as a Solution for Big Data 0 2020 - NIH EA - Persisent Link
- NIH EA - dated 1995 - 2003
- NIH Accessiblity - Section 508: The NIH Office of the Chief Information Officer (OCIO) is responsible for the oversight of the Section 508 program at NIH. Section 508 refers to an amendment that was added to an existing law – the Rehabilitation Act of 1973 that requires all electronic and information technology (EIT) that is developed, procured, maintained, or used by the Federal government be accessible to people with disabilities.
- NIH - EA - Key Behaviours: . . . .
- NIH EA Search Results

TO DO

- [ ] : Enterprise Reference Architectures
- [ ] : Enterprise Application Reference Architecture
- [ ] : c4model - system ___c___ontext, ___c___container, c___omponent, ___c___ode
- [ ] : Scope/Audience : identify scope (Gartner How to Doc App Arch recommendation)
- [ ] - Developer Experience : How can archtecture improve Developer Experience (modern technology, known expecations, concern resolution channels, . . .)
- [ ] - Interrogatives What-Date, How-Function, Where-Network, Who-People, When-Time, Why-Motivation. Refer to DoD AF
- [ ] - Applicability to Agile - from Gartner 2021 - How to Document Application Architecture - G00750245 *Which framework you use matters*

less than using a framework in the first place. Bear in mind that the more heavyweight and prescriptive the framework is, the less well-suited it is to agile development. If your internal processes call for you to create detailed architecture documentation in the early phases of a development effort, then they're not agile.

- [ ] GC Cloud Guidance : NO longer "Cloud First", but Cloud ???
- [ ] GC Digital Standards

Architecture: (from Gartner)

- Concepts:,
- Principles:
- Rules:
- Patterns:
- Interfaces:
- Standards:
- Reference Models:

Software / Code Quality: (from Gitlab) : code quality, SAST & container scanning

Architecture: [(from Google) - functional Requirements - non-functional requirements (some call these quality requirements) - constraints

@Work: Directives, Guidelines, Standards, Processes, Procedures, . . .

Enterprise Overview

Guidelines

Security : Shift-Left

Engage Security early. Inform security early of the technology you are planning on using, and how you are address common security requirements (identity, access, monitoring, auditing, patching, . . .). There are many other security families, and including security and recording joint, collaborative decisions is important in avoiding the discovery of major security risks later in the project.

System Architecture

Loosely Coupled

Resources: Domain Driven Design : Domain Driven Design can help decompose a system into loosely coupled components: how to draw boundaries between services, how to decide whether some logic belongs to one service or another, and how domain-driven design can help us make those decisions.

Design Methodolgoy??

User Experience

Embrace GC User Centric Design - Nielsen Norman Group - Maturity Model

Maturity Level	Description
Absent	UX is ignored or nonexistent.
Limited	UX work is rare, done haphazardly, and lacking importance.
Emergent	The UX work is functional and promising but done inconsistently and inefficiently.
Structured	The organization has semi-systematic UX-related methodology that is widespread, but with varying degrees of effectiveness and efficiency.
Integrated	UX work is comprehensive, effective, and pervasive.
User-Driven	Dedication to UX at all levels leads to deep insights and exceptional user-centered–design outcomes.

4 high-level factors that contribute to the organization's UX maturity NNG - UX Factors to Success 1. Strategy: UX leadership, planning, and resource prioritization (vision, planning, budget) 1. Culture: Knowledge about and attitudes towards UX, as well as cultivating UX careers and practitioners' growth. (awareness, appreciation, competencys, adaptability) 1. Process: Systematic, efficient use of UX research and design methods (methods, collaboration, consistency) 1. Outcomes: Intentional definition of goals and measurement of the results produced by UX work (impact of design, measurement)

UX Design

- follow GC User Centric Design
- see deisgn mothers for Ux - on CW - UK 18F Methods - referenced by GC standards - 18f technoloyg and design company - USA
- http://uxdesign.uw.edu/ j -Usability.gov - User Centric Design

Observability

Observability: This is an entire field of its own, and has gained momentum with microservices.
- Trace - Log - Montor ???

Reference Architecture

CNA - AWS : Cloud Native

https://aws.amazon.com/architecture/ - Analytics and Big Data
- Compute and HPC (high performance computer) - Databases - Machine Learning

data bricks : cloud hosted data engineering, data analytics, data science (core is Apache Spark)

- Snowflake - Modern Reference Architecture for Applications
- Snowflake : cloud-hosted, relational database for data warehouses. (Markets as replacement for Hadoop) - ref: blog
- https://www.snowflake.com/blog/
- https://www.snowflake.com/blog/modern-reference-architectures-for-application-builders/
- Snowflake - Choosing Open Wisely
- *we strongly believe in API-driven access to data*

Example Problems

Enterprise Search

Problem: Enterprise Search has access to all corporate information. Display search results, and provide information to the user must respect security and privacy concerns. Applications with local access-business logic, and big-data platforms are difficult to expose to enterprise search. Challenges: 1. Granularity: field-level access, crawl PDFs (and other filetypes), … 1. Discoverability: search 1. Search Relevance: 1. Precision : Precision is the percentage of documents in the result set that are relevant. Find relevant documents, with very few irrelevant documents. 1. Recall : Recall is the percentage of relevant documents that are returned in the result set. Recall means the number of documents retrieved that are relevant, divided by the number of total relevant documents. 1. Saved Search : Ability of saving a search and working through the search results, refining search (eDiscovery)

? What is ElasticSI (elastic search + ??) - cloud based search : https://elastic-si.com/

Data Availability

Problem: Corporate solutions (ERP, HR, S-app, Collab-platforms) have silos of data which is unavailable for easy use by other applications. Guideline: "Application data must be made available to other applications in a controlled fashion". - Controlled Fashion : respecting IM and security concerns. Exposing data in a modern manner (distributed architecture, API).

Framework for Data Governance

Problem: Data governance, business and technical processes to provide a version of truth of the system, or, system of record. Example: Entity Resolves / Unresolves. The business-activity is unrestrained, and the application that performs the resolution action does not generate meaningful data such that a reliable API can be defined to identify a resource. The system of record is unaware of these activities. The SoR does not have API to identity the identity of the entity, nor, a means to notify on changes to the core entity's identity. Impact: An external application consuming this information has developed business logic to arbitrate entities, and, has nightly jobs to re-validate the integrity of its business logic. Resolutions: Identify a system of record for entities and ensure the business logic for this responsibility is contained within this system. Technologies that can help: Data governance, Master Data Management. Software Architecture Patterns that can help:
- Bounded Context - Domain Driven Design : DDD model of the entity (our Entity), events on the entity (resolve / unresolve) and aggregates which are only permitted to contain references to the root of the entity. Challenge: Unsure how MDM can solve Analytics Platform resolve/unresolve process.

Distributed Multi-Security-Zone Business Process

Problem: We are starting to use SaaS application providers, however, due to privacy and security concerns we cannot put sensitive information in the SaaS application. As a result, business process will involve using systems on both the high and low security zones. Business users will have to learn to complete processes using tokenized, masked and redacted information. IT will have to provide technology to move and transform the data between these systems. This design constraint will impact the user experience. IT's role is to implement technology to facilitate these multi-zone business processes, while reamining loosely coupled with the SaaS solution (important as the SaaS solution is upgraded regularly beyond the control of the business or IT). Guideline:
- TBD: Patterns include EDA, distributed architecture Decisions: - business logic will not reside on the low side. All integration business logic will be on the high-side.

Identity Quote : Okta?

Terms

- Application:
- System:
- Solution:
- Reference Architecture : A reference architecture is a document or set of documents that provides recommended structures and integrations of

IT products and services to form a solution. The reference architecture embodies accepted industry best practices, typically suggesting the optimal delivery method for specific technologies. A reference architecture in the field of software architecture or enterprise architecture provides a template solution for an architecture for a particular domain. A lower level one might demonstrate the interactions of procedures (or methods) within a computer program defined to perform a very specific task.

- A Reference Architecture is created by capturing the essentials of existing architectures and by taking into account future needs and opportunities, ranging from specific technologies, to patterns to business models and market segments. Reference: THe Gaudi Project - Netherlands

CNA

- CNA: Cloud-native architecture and technologies are an approach to designing, constructing, and operating workloads that are built in the cloud and take full advantage of the cloud computing model.
- Kubernetes is a platform to manage, host, scale, and deploy containers.
- Containers are a portable way of packaging and running code. They are well suited to the microservices pattern, where each microservice can run in its own container.

References

Gartner

GC

1. Directive on Service & Digital - Appendix B - Mandatory Procedures

Wikipedia

1. View Model
2. USA Federated Enterprise Architecture : Viewpoints of Enterprise, Information, Functional, Physical, Engineering, Technology. 6 Reference Models : Performance, Business, Data, Application, Infrastructure, Security.
3. USA NIST Enteprise Architecture Model - 1980's no longer relevant

Documentation

- Google DORA State of DevOps - 2021
- IEEE Software Documentation Issues Unveiled - ISBN 978-1-7281-0869-8 - 2019
- IEEE The Value of Software Quality - ISBN 978-1-4799-7198-5 - 2014

Standards Bodies

1. [IEEE 42010] - Systems and Software Engineering - Architecture Description (2011 $150)

- The Journal of Systems and Software - Overview of Architecture Description - Documenting Decisions - IEEE 42010

Frameworks

TOGAF Archimate Zachman

USA

DoD Architecture Framework - massive framework - notes on USA Gov - coarse grained API … (API for large data, …)

Templates

This section addresses the lack of standards, templates and clear requirements for documentation. - Integration COE : standard for schemas (API and events) and knowledge of consumers/subscribers to these integration points.

Goal

- *At the end of the day, I want my artifacts to be enduring. My goal is to create a prescriptive, semi-formal architectural description that can be used as a basis for setting department priorities, parallelizing development, [managing] legacy migration, etc.*

Thoughts - Describe TRB / AWG differences

- key - TRB is agile - share early, quickly, often and informally…get guidance…

Unstructured Knowledge Sharing: - team-based knowledge sharing meetings where the purpose is to share and discuss (some TRB meetings have this as the key-goal) - open office hours for any questions - lunch & learn presentation series
- show & tell / demonstrations
- peer learning groups : learning (small, L&D type activities), guided learning

Footnotes - Test

Here's a sentence with a footnote. [1]

Here is a simple footnote[2]. With some additional text after it.

[1] This is the footnote.
[2] My reference.

It is reference links emphasis tutorials. It is reference links emphasis tutorials.