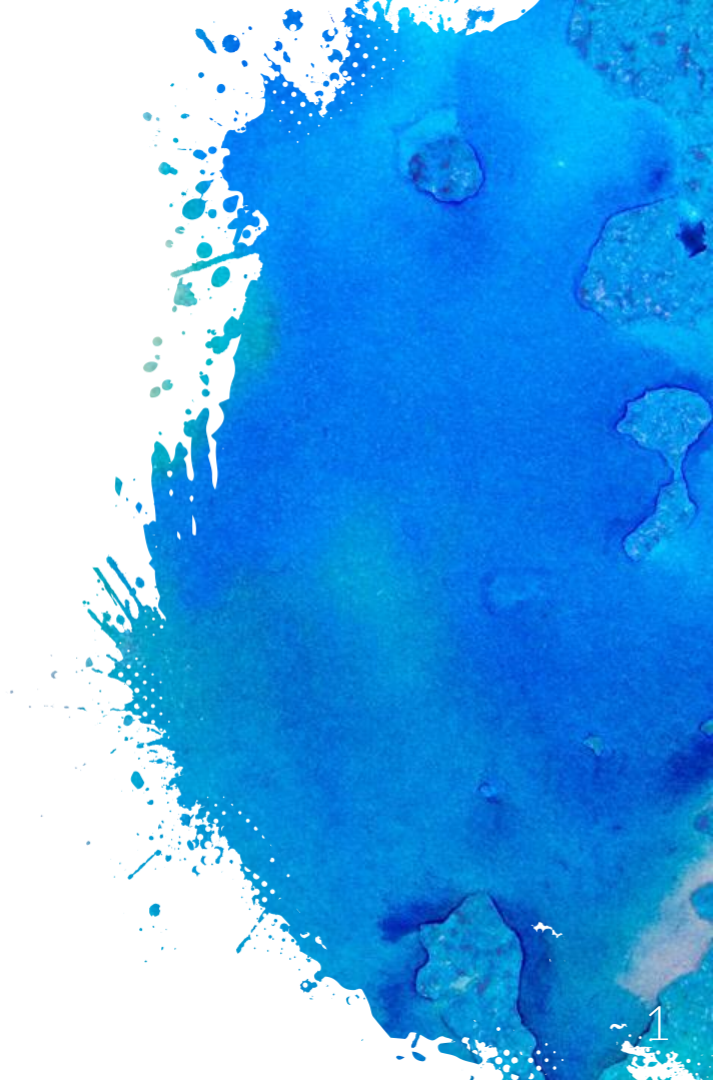
FHIR Architecture 101

30/03/22



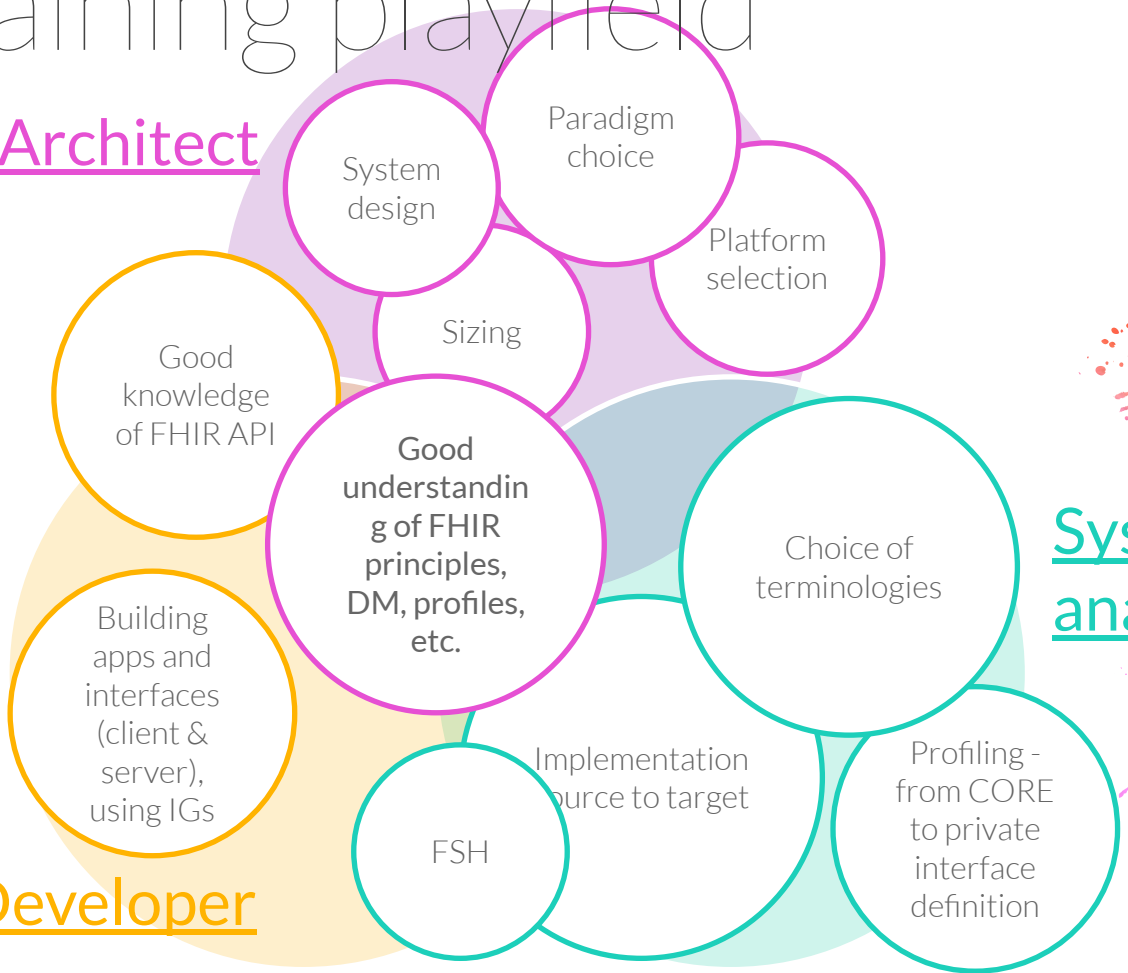
TOC

- Intro to FHIR
- Training playfield
- Architecture examples
- General guidelines
- Questions

Fast Healthcare Interoperability Resources

A standard describing **data models** (known as "resources") and an **API** for electronic exchange of healthcare information

Training playfield



Our approach – learn by example

Scenario

- Present scenario (as close to your use-cases as possible) including architecture diagaram
- Discuss solution approach with focus on specific categories

Paradigm

FHIR interoperability paradigm as outline by HL7 (RESTful API, Messaging, Documents, Services, Database / Persistent Storage, Subscriptions Framework). We'll also touch upon servers, façades, etc.

Tools

Tools and platforms you can use

Secuirty

Security aspects including both standard and proprietary approaches, authentication, authorization, permissions enforcement, etc.

DQA

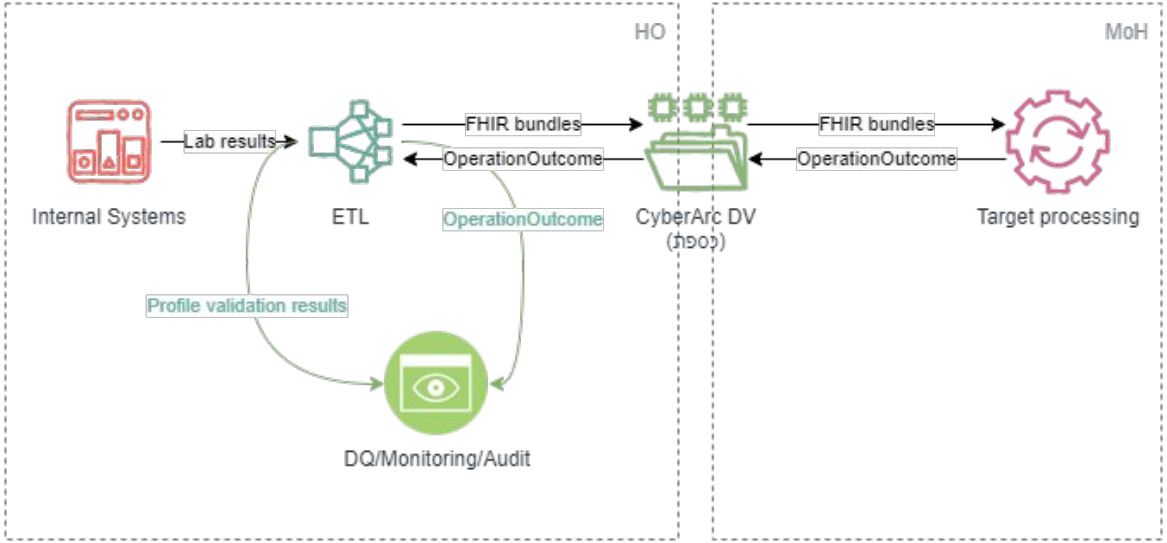
Data quality monitoring and cleanup aspects and strategies

I meant that façade

Scenarios

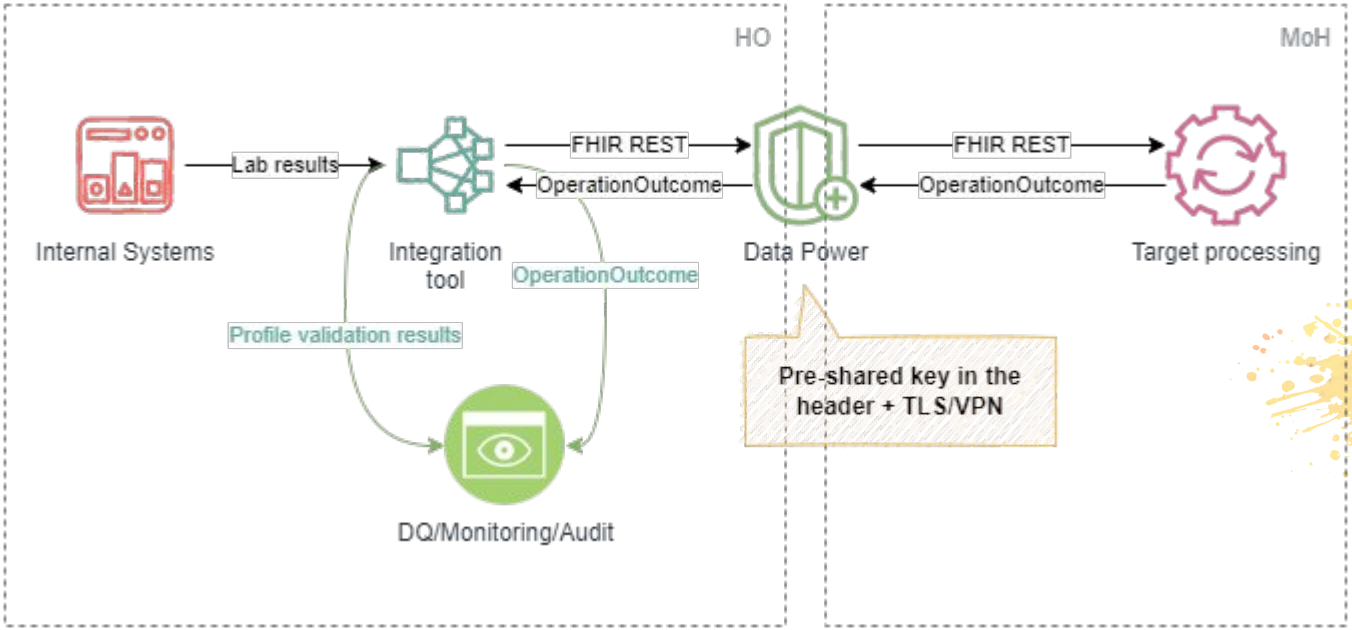
	Scenario	Description
1	Sending COVID test to MoH	Generate FHIR bundles via your favorite integration tool, run profile validation on them and put them in safe. Collect operation outcome files from save, pace them and feed results into your DQA dashboard.
2	Sending COVID test to MoH #2	Generate FHIR bundles via your favorite integration tool and POST them to REST API. Receive operation outcome response from API call and feed results into your DQA dashboard.
3	Get data from HMO - CoverageEligibility	Send CoverageElegibilityRequest and get CoverageElegibilityResponse or OperationOutcome and handle errors
4	Provide CoverageElegibility to hospital	Recieve CoverageElegibilityRequest, fetch data from internal systems and send CoverageElegibilityResponse or OperationOutcome
5	Provide CoverageElegibility to hospital #2	Recieve CoverageElegibilityRequest, fetch data from internal systems and send CoverageElegibilityResponse or OperationOutcome
6	Provide CoverageElegibility to hospital #3	Accept extended operation API call to check coverage eligibility, fetch data from internal systems and return CoverageElegibilityResponse or OperationOutcome
7	Accept ServiceRequest for labs & provide results data	Accept ServiceRequest in FHIR and route to backend system via internal mechanism. Also handle requests for DiagnosticReports/Observations and fetch them from the backend system. Note that you'll need to handle search, includes/revincludes, security and terminology mapping
8	Accept ServiceRequest for labs & provide results data #2	Accept ServiceRequest in FHIR via message and process it asynchronously. Once the results are available send back DiagnosticReports/Observations via message. Mind security and terminology
9	Accept ServiceRequest for labs & provide results data #3	Accept ServiceRequest in FHIR via message and process it asynchronously. Once the results are available push DiagnosticReports/Observations into full FHIR server that can be queried. A CQRS example
10	Get vaccination updates	Get Immunization from MoH (corona, schools, etc.) via REST and push them into backend system. You'll have to handle transactions, terminology mapping, validation yourself. You should also consider cases where you'll receive unexpected data. You'll also need to address identities reconciliation
11	Get vaccination updates #2	Get Immunization from MoH (corona, schools, etc.) via REST into full FHIR server, subscribe to changes on the server and stream them to your backend systems. You'll also need to address identities reconciliation
12	SMART on FHIR patient app	Populate FHIR server with patient data and provided it to SMART on FHIR apps

Sending COVID tests to MoH



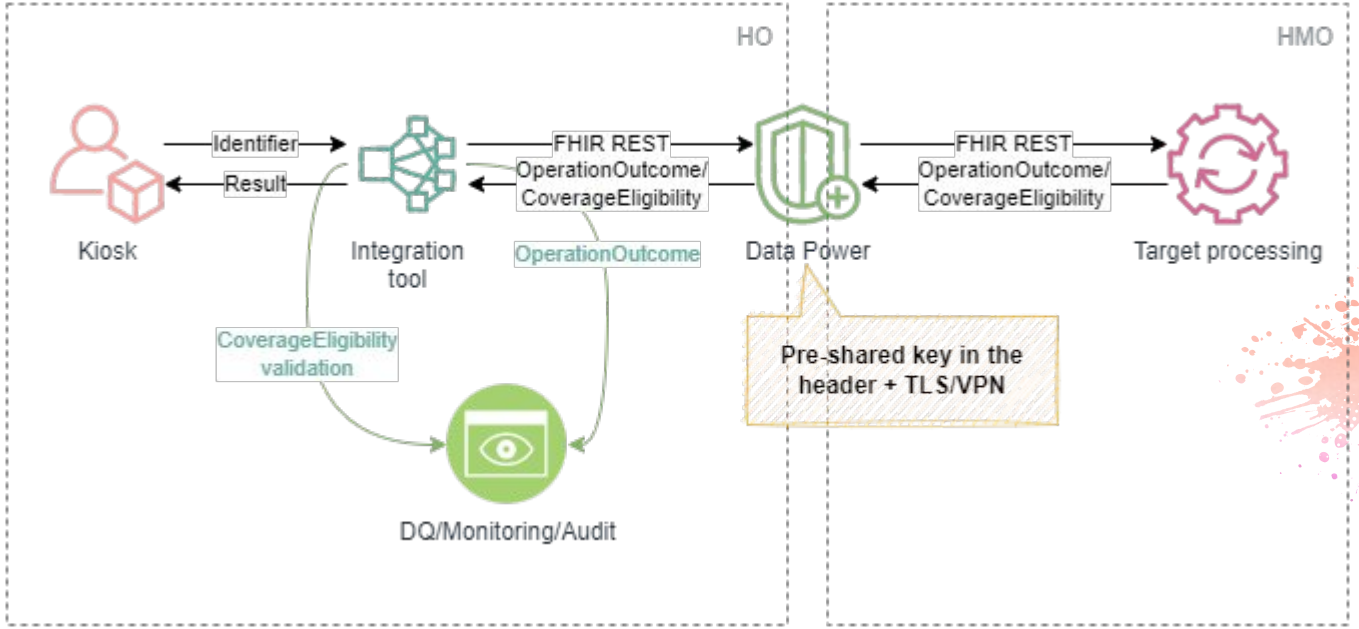
Description	Paradigm	Tools	DQA	Security
Generate FHIR bundles via your favorite integration tool, run profile validation on them and put them in safe. Collect operation outcome files from save, pace them and feed results into your DQA dashboard.	Persistent storage (Client)	Integration tool, preferably with FHIR support (Tibco BW6 with plugin, Talend, nifi with HAPI client, DBT, etc.), code (python, .net, java, node, etc.), HL7 Java validator	Run profile validation on bundles using HL7 Java validator. Route exceptions to your quality team. Collect OperationOutcome files from MoH, parse and route exceptions to your quality team. You should also send counts, but that doesn't have to be in FHIR	Implicit - provided by safes

Sending COVID tests to MoH #2



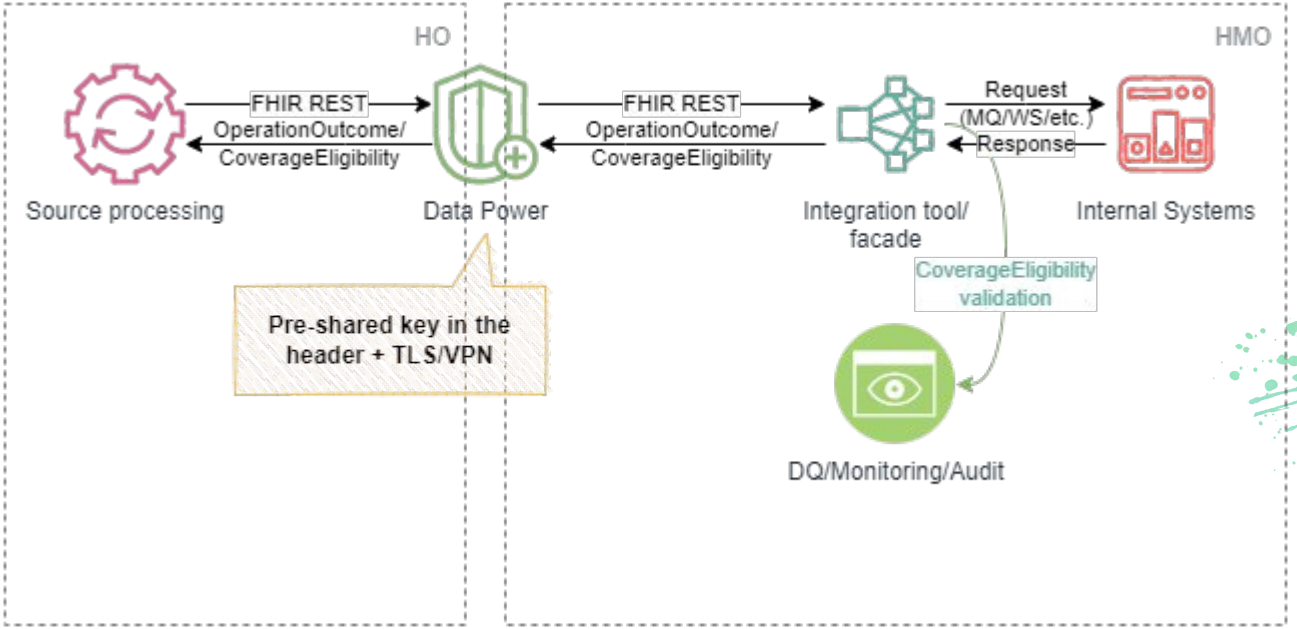
Description	Paradigm	Tools	DQA	Security
Generate FHIR bundles via your favorite integration tool and POST them to REST API. Receive operation outcome response from API call and feed results into your DQA dashboard.	REST API (Client)	Integration tool, preferably with FHIR support (Tibco BW6 with plugin, Talend, etc.), code (python, .net, java, node, etc.), HL7 Java or another validator	It's advised to run profile validation on bundles before POSTing them. Receive OperationOutcome response from MoH, parse and route exceptions to your quality team	Pre-shared key in the header + TLS/VPN

Get data from HMO - CoverageEligibility



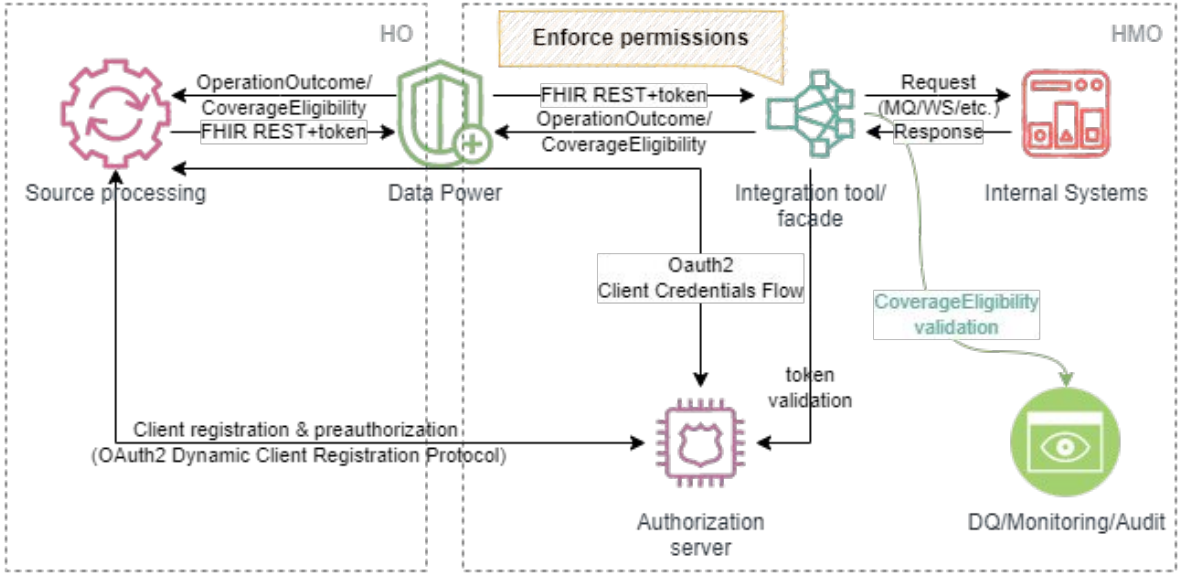
Description	Paradigm	Tools	DQA	Security
Send CoverageElegibilityRequest and get CoverageElegibilityResponse or OperationOutcome and handle errors	REST API (Client)	Integration tool, preferably with FHIR support (Tibco BW6 with plugin, Talend, etc.), code (python, .net, java, node, etc.)	You can run full profile validation on CoverageElegibilityResponse, but since there is custom business logic required to process response, you might do all validation there. You will need to handle OperationOutcome with errors	Pre-shared key in the header + TLS/VPN

Provide CoverageElegibility to hospital



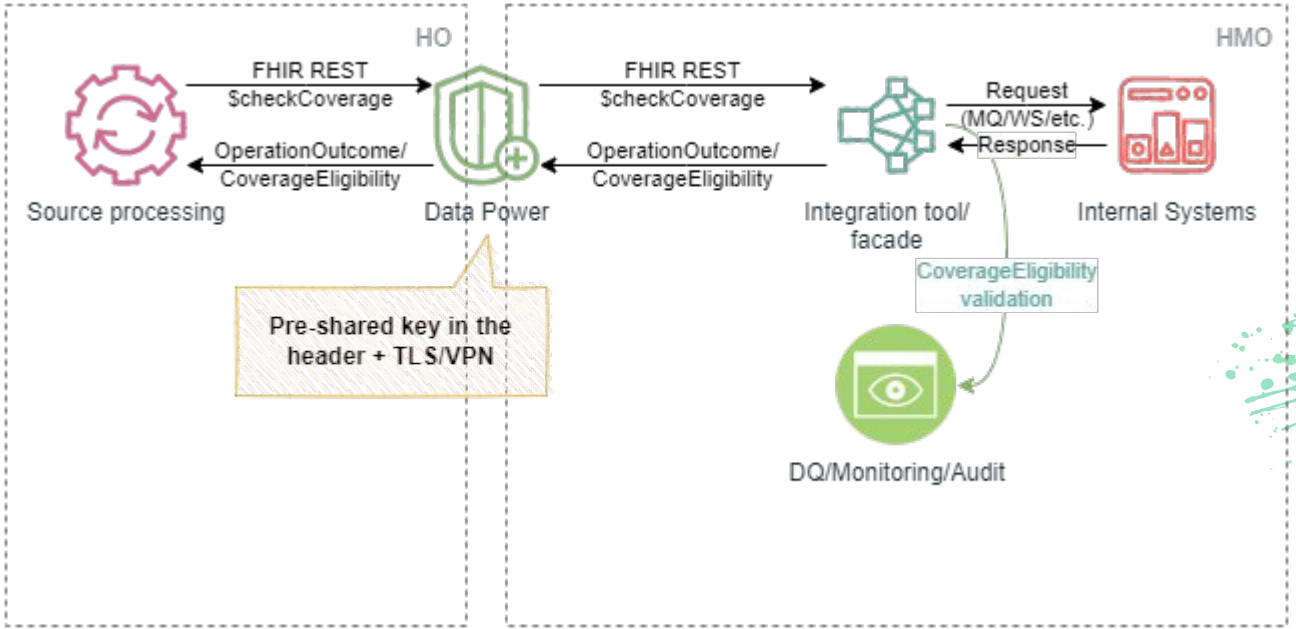
Description	Paradigm	Tools	DQA	Security
Receive CoverageElegibilityRequest, fetch data from internal systems and send CoverageElegibilityResponse or OperationOutcome	REST API (façade)	Tibco BW6 with plugin, Hapi or any other commercial façade options, code	You can run full profile validation on CoverageElegibilityRequest, but most of the validation is business logic, so it makes more sense to do custom validation instead	Pre-shared key in the header + TLS/VPN

Provide CoverageElegibility to hospital #2



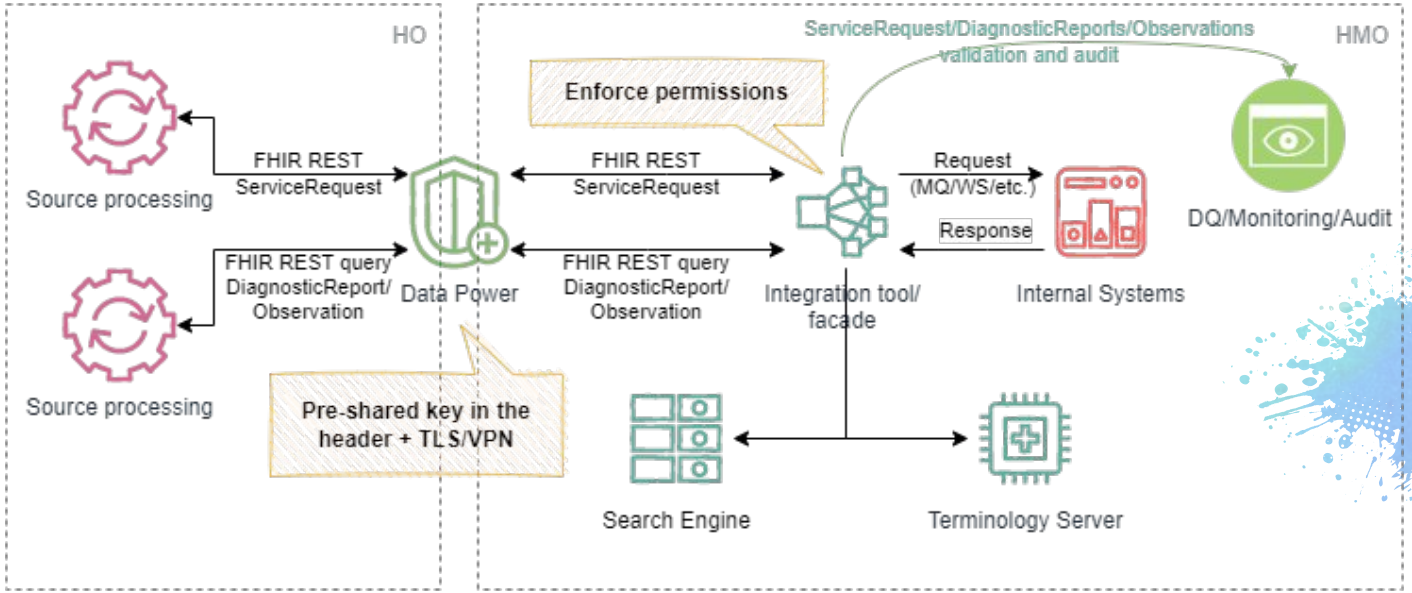
Description	Paradigm	Tools	DQA	Security
Receive CoverageElegibilityRequest, fetch data from internal systems and send CoverageElegibilityResponse or OperationOutcome	REST API (façade)	Tibco BW6 with plugin, Hapi or any other commercial façade options, code	You can run full profile validation on CoverageElegibilityRequest, but most of the validation is business logic, so it makes more sense to do custom validation instead	Authenticate using SMART Backend Services profile Requires preauthorization (registering of service, keys exchange) and use of client credentials grant flow. FHIR server is still responsible for validating access token and enforcing permissions

Provide CoverageElegibility to hospital #3



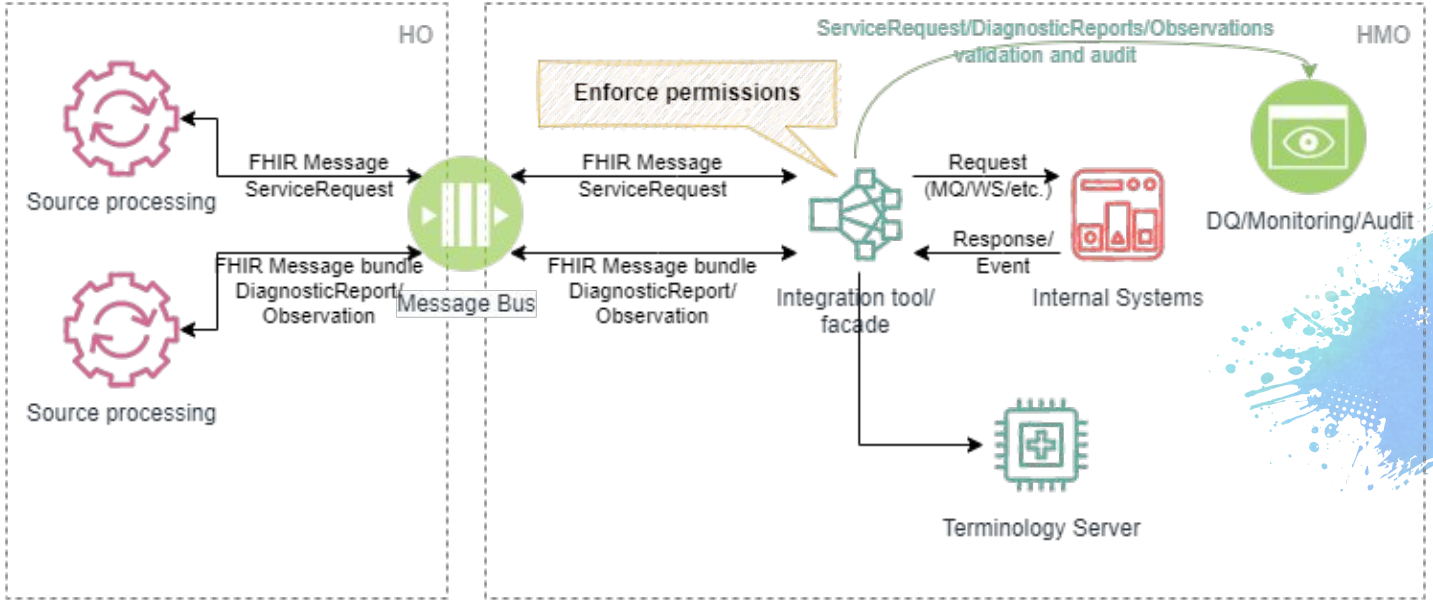
Description	Paradigm	Tools	DQA	Security
Accept extended operation API call to check coverage eligibility, fetch data from internal systems and return CoverageElegibilityResponse or OperationOutcome	REST API (façade)	Tibco BW6 with plugin, Hapi or any other commercial façade options, code	You can run full profile validation on CoverageElegibilityRequest, but most of the validation is business logic, so it makes more sense to do custom validation instead	Pre-shared key in the header + TLS/VPN

Accept ServiceRequest for labs & provide results data



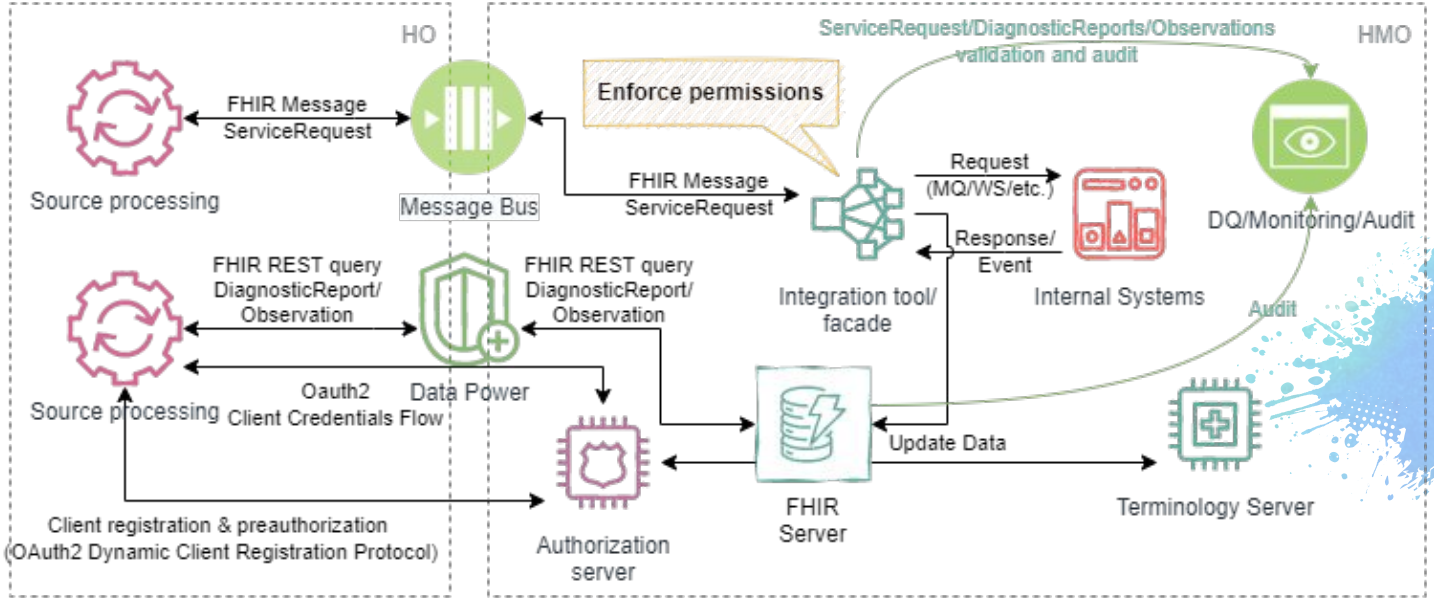
Description	Paradigm	Tools	DQA	Security
Accept ServiceRequest in FHIR and route to backend system via internal mechanism. Also handle requests for DiagnosticReports/Observations and fetch them from the backend system. Note that you'll need to handle search, includes/revincludes, security and terminology mapping	REST API (façade)	Tibco BW6 with plugin, Hapi or any other commercial façade options, code	You can run full profile validation on ServiceRequest, but most of the validation is business logic, so it makes more sense to do custom validation instead. You should also do some sort of validation (but not necessarily full profile) on DiagnosticReports/Observations to make sure they are valid before serving them	Pre-shared key in the header + TLS/VPN

Accept ServiceRequest for labs & provide results data #2



Description	Paradigm	Tools	DQA	Security
Accept ServiceRequest in FHIR via message and process it asynchronously. Once the results are available send back DiagnosticReports/Observations via message. Mind security and terminology	Messaging	Tibco EMS with plugin, Kafka with code, etc.	You can run full profile validation on ServiceRequest, but most of the validation is business logic, so it makes more sense to do custom validation instead. You should also do some sort of validation (but not necessarily full profile) on DiagnosticReports/Observations to make sure they are valid before serving them	Implicit - provided by messaging infrastructure & VPN. Make sure you only send authorized data.

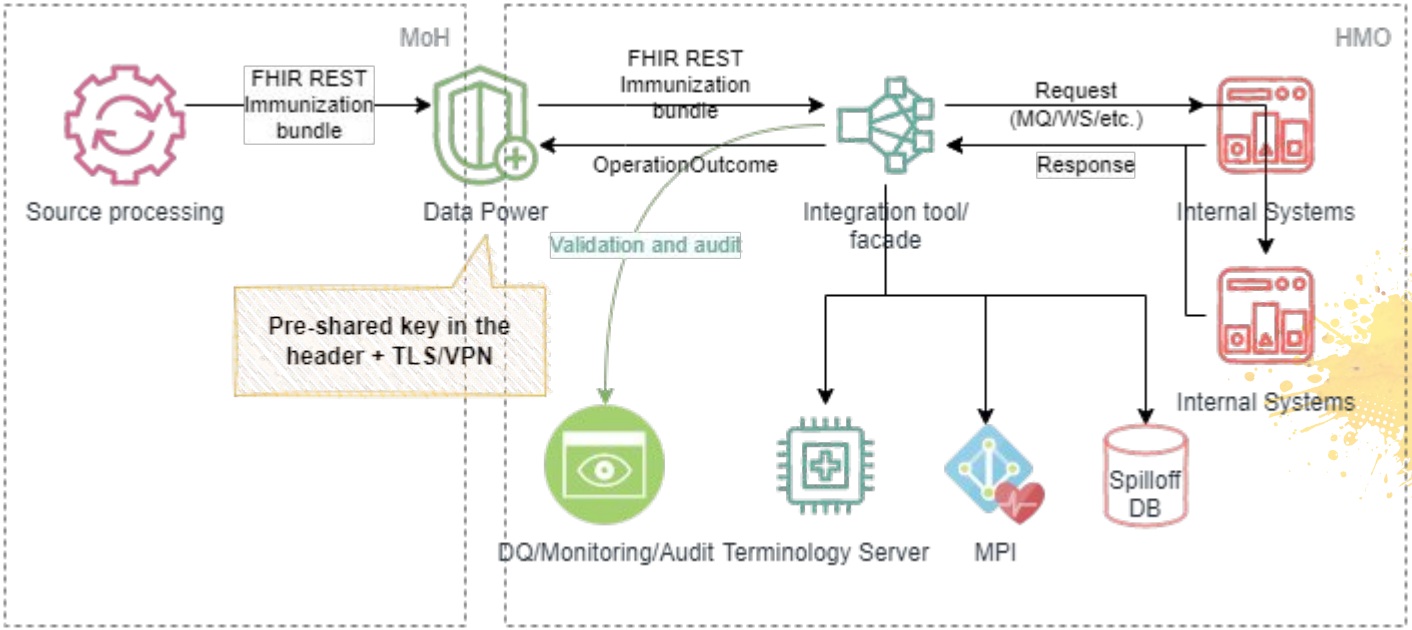
Accept ServiceRequest for labs & provide results data #3



CQRS Example

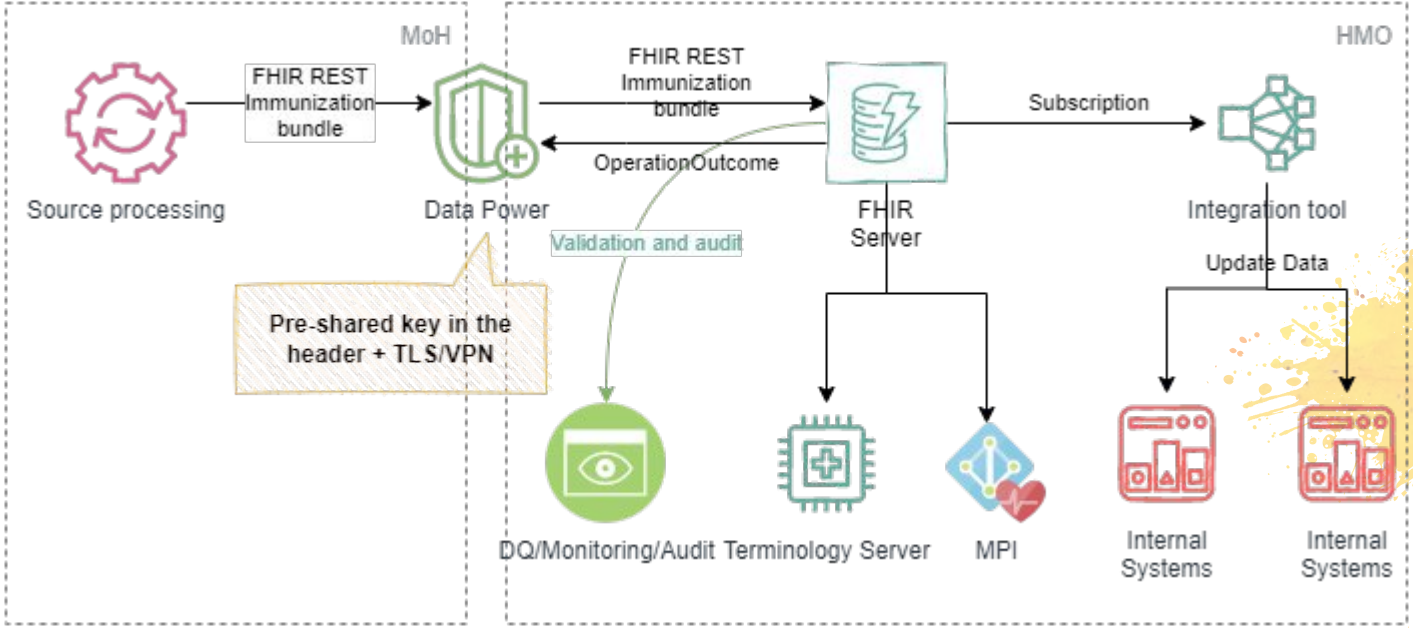
Description	Paradigm	Tools	DQA	Security
Accept ServiceRequest in FHIR via message and process it asynchronously. Once the results are available push DiagnosticReports/Observations into full FHIR server that can be queried. A CQRS example	Messaging + REST API (server)	Inbound - Tibco EMS with plugin, Kafka with code, etc. Outbound - Full FHIR server (HAPI, any of the commercial options)	You can run full profile validation on ServiceRequest, but most of the validation is business logic, so it makes more sense to do custom validation instead. You should also enable profile validation on DiagnosticReports/Observations in FHIR server	Inbound - Implicit - provided by messaging infrastructure & VPN. Outbound - using any authentication/authorization mechanism (i.e. - SMART backend services) but FHIR server must enforce permissions

Get vaccination updates



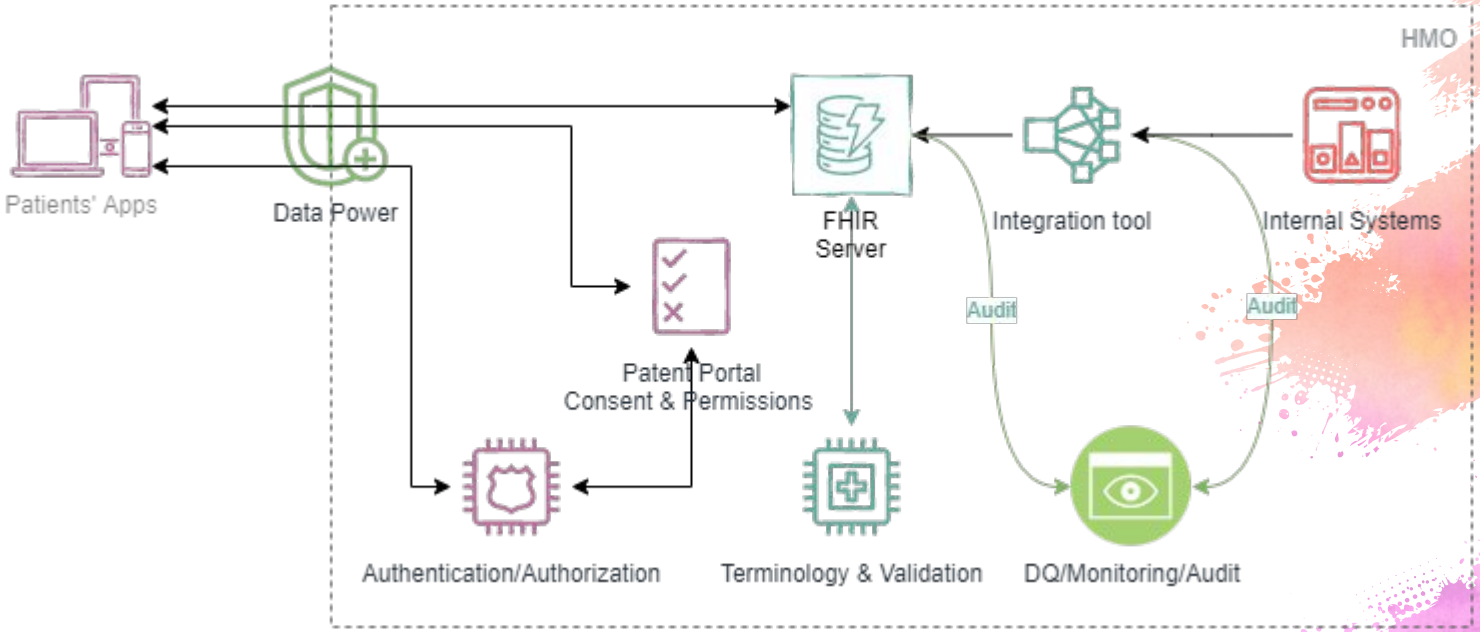
Get Immunization from MoH (corona, schools, etc.) via REST and push them into backend system. You'll have to handle transactions, terminology mapping, validation yourself. You should also consider cases where you'll receive unexpected data. You'll also need to address identities reconciliation	REST API (façade)	Tibco BW6 with plugin, Hapi or any other commercial façade options, code	You should do full profile validation on received data and surface errors to your DQA team and send OperationOutcome back to MoH. If online validation is too resource-intensive, you can do it asynchronously. You'll also need to handle identity reconciliation (as you might get different business identifiers).	Pre-shared key in the header + TLS/VPN

Get vaccination updates #2



Get Immunization from MoH (corona, schools, etc.) via REST into full FHIR server, subscribe to changes on the server and stream them to your backend systems. You'll also need to address identities reconciliation	REST API (server) & Subscriptions	Full FHIR server (HAPI, any of the commercial options)	You should enable full profile validation on received data and surface errors to your DQA team and send OperationOutcome back to MoH. If online validation is too resource intensive, you can do it asynchronously. You'll also need to handle identity reconciliation (as you might get different business identifiers). some servers provide support for it, and you can also use a MPI FHIR server.	Pre-shared key in the header + TLS/VPN
Description	Paradigm	Tools	DQA	Security

SMART on FHIR patient app



Description	Paradigm	Tools	DQA	Security
Populate FHIR server with patient data and provided it to SMART on FHIR apps	REST API (server)	Full FHIR server (HAPI, any of the commercial options)	Resources in your FHIR server should be compliant with IL-CORE profiles and you should also validate incoming resources	Full SMART on FHIR app profile, authentication of user, consent for access and permissions enforcement

General Guidelines

Laziness is a key to interoperability!

(avoid doing anything "special“)

- Prefer REST
- Permissive interfaces
- Don't lock yourself into vertical use-case
- Each paradigm can change security model


Thanks!
Any questions?