BUSINESS REQUIREMENTS GATHERING FOR DESIGNING AN API-BASED SERVICE

A) HOW IT'S CURRENTLY DONE:

{@Manix: place holder for the current ETL way}

B) The API way

B.1. API Name:

e.g. locationService

B.2. Version:

If we are collecting info for an RFC, indicate the API version that needs to be updated

B.3. SOURCE SYSTEMS

. Name the source system(s) that will be consuming this API

. Systems specifics: home grown, technology stack or COTS

. Source system is on-premise or is cloud based?

. Network access: Can your source system already access an API in the cloud? If not what are the pre-requisites?

B.4. Targeted Resources

What are your targeted systems in this integration (not sure if we can expect this from the Business SMEs)

B.5. List of Methods

what are the operations (methods) this new Service needs to support?

e.g. FOr a Location service:

. AddLocation

. UpdateLocation

. Remove Location

. Find Location

. LocationList

B.6. Authentication mechanism

one-way, two-way SSL, OAuth, SAML?

If unsure, Define your security requirements in business terms

B.7. API Processing Strategy

Synchronous or Asynchronous?

i.e. when making a request to the service will the caller be waiting for a response (sync) or not waiting (async)

and be notified a bit later that a response is available.

B.8. Data format

what data format can you send and what data format can you receive?

e.g. JSON, XML, txt, csv, other legacy formats...

B.9. MAPPINGS

For each API method (e.g. 5 of them as above),

describe the data you will sending to the service method and what data you are expecting in response.

. Object name, attributes name, attributes data-types

. describe object hierarchy: parent-child, 1-to-1, 1-to-many, many-to-1

. qualify attributes as mandatory or optional

. Define acceptable values for certain fields (Enums)

. Specify how your system will handle optional attributes:

. ignore attribute

. send blank attribute value

. send null value for attribute

. For numerical attributes, define int, long, double or if sent as string "123.45"

. For Date & time attributes: specify format, UTC format, time zone if applicable

. Include sample data (as companion files, not in this doc)

B.10. Estimated (wish) release schedule for API

in test, in UAT, in production

B.11.Usage, Load, throughput

. Estimated use/load (frequency): how many times per day/hour/minute will your system be making a request.

Provide granularity per method ideally.

. Estimated Request payload size, response payload size?

How much data in each request, how much data in each response?

B.12. SLAs

. SLA for this service? possibly method by method

. Estimate Service Timeout

B.13. EXCEPTION HANDLING

For a SYNCHRONOUS behavior: when an exception occurs, what is your system expecting as a response to indicate that error?

. HTTP 500 code

. HTTP Error Json object

. CloudHub notification,

. an exception manifest in a database, a file, an email or an sms, other...

B.14. TRANSACTIONAL BEHAVIOR/REVERSALS

. How would you like to roll back/undo/reverse operation in case a transaction results in data corruption or dirty data state?

B.15 EXCEPTION HANDLING

. Client-side expected error handling:

. HTTP response with 500,

. with Error Json object,

. Define Last recourse (Dead Letter Queue) behavior when all attempts have failed: Error log entry, exception manifest, Email, SMS, other

B.16. ASYNC BEHAVIOR (for Asynchronous integrations only)

. When your system invokes the API, does it expect an immediate ACK (from the API) for fire-and-forget type of processing?

i.e. the API notifies the source system that it receives the request (and its payload)

. What is the format of this ACK (JSON object)? or other format (XML...)

. Once the response has been processed, what callback mechanism is the source system able to handle? REST, SOAP, EMAIL, SMS, Message Queue, other?

. What is the callback response format (JSON, text, other)?

. If client supports REST callback (ideal), what is the URL, REST verb and specifics of this callback? (query strings params or http headers,...)

B.17 AVAILABILITY - FALLBACK

. Expected availability (24/7 or certain time windows)

. Impact on business if single point of failure

. Acceptable downtimes (minutes per month)

. System maintenance downtimes

. If API/Service is unavailable what would be your alternate course of action?

B.18 MONITORING & TRACEABILITY

. Would your system be capable of sending to the service some HTTP custom headers for traceability purpose?

e.g. SourceSystem="my-system-name", TargetSystem if applicable "that-other-system", transactionID (a UUID).

B.19. WORKFLOWS

Textual description of:

. the overall workflow between your system and the service API. 1 diagram per service method ideally (if their behaviors are different).

B.20.SPECIAL HANDLING

If Client system does not support REST but can only send a file or uses sockets or else, describe your constraints and pre-processing requirements.

Other types of constraints worth communicating.

B.21. User Stories / Use cases

. Describe the business requirements

. Describe the Functional requirements: i.e. divide the business requirements in distinct operations

Each operation is likely to transalate to a specific API/Service method

. Provide a happy path and unhappy path scenario for each service method.

. Method #1: AddLocation

describe the use case in plain text, include unhappy path as well

. Method #2: UpdateLocation

...

B.22. Assumptions, Business and Functional

B.23.Quality Assurance

. End-to-end Tests Acceptance criteria: at least 2 per method: one happy path, one unhappy path

B.24. Analytics: define basic analytics requirements

e.g. Number of API invocations, % of service errors, Usage over time...

B.25. Management, Monitoring and Maintenance

. define ports to monitor on source system

. define list of business events to record through the integration (e.g. number of Locations created, number of locations updated, $amounts if sales related...)

. define list of Ops Team recipients for email alerts, for regular analytics reports...

B.26. Additional Information

Define other specifics of your business not yet captured in this document

e.g. future system changes, cyclical system updates...