Sourcing Services Design

Including service model, technical approach and technical requirements specification.

[**Overview**](#_kvfaln3te9pt) **2**

[**Context**](#_3wyxxvahui88) **2**

[Baseline](#_88hdn1dw0q5g) 4

[**Principles**](#_67s1965ewji5) **4**

[**Service map (context)**](#_qfm1m5arqnwi) **5**

[**Technical Specification**](#_uu06r4bhu6mb) **6**

[Description](#_qzp9qerkbpdl) 7

[Internet layer](#_jcpwmj3k3zqz) 7

[Process layer](#_tyb7trxwipcl) 8

[Record layer](#_kfglofvw3nt3) 9

[Directives](#_c65jl97x9we7) 9

[Common directives](#_wsqv3nscimxi) 10

[Internet layer directives](#_clqd5gb9pwsh) 10

[Process layer directives](#_bdk5uuqtt8sq) 10

[Records layer directives](#_bn9sqnr1a7yo) 11

[**Solution options**](#_6h89x9lytewv) **12**

[**Transition Approach**](#_uqgpj6fux0bc) **13**

[**Evidence**](#_besez5nhsstt) **14**

[Integration approach](#_q57d27cuw4y7) 14

[Gartner](#_4t73hh10aor) 14

[Targeting solution components](#_co6c1iokoayq) 14

[Agility](#_360vupxabry1) 15

# Overview

This is a high level service design document outlining scope and intent for sourcing solutions, and establishing context also for call-off and contract management activities.

It specifies a technical environment to ensure change vehicles are coordinated, and build-versus-buy decisions are in context.

# Context

There is one giant ‘journey’ for an agreement[[1]](#footnote-1) that CCS users may want to consume, from discovery, through framework/DPS/contract design and procurement, to marketing and call-off through to management and reporting.



Fig one - simplified journey of an ‘agreement’

1. We need to design solutions to allow for this journey, but not expect a single solution option (especially purchased product) to resolve it all (Fig 2 below)
2. Procurement software market is mature - products are evolved for many of the problems in the discipline although integration is not strong [Gartner hype cycle and others - see Evidence].
3. However, Procurement software market is in flux and we will be in transition between products every few years [Gartner hype cycle and others - see Evidence]
4. Procurement (and ERP more generally) is moving to a dis-aggregated rather than a monolithic model [Gartner digital ERP - see Evidence]
5. Gov tech procurement standards [[TCOP](https://www.gov.uk/government/publications/technology-code-of-practice/technology-code-of-practice)] require us to, amongst other things:
   1. deliver based on end user needs
   2. maintain ownership of our data
   3. use open standards
   4. share and reuse technology
   5. be open
6. Customer departments have their own products
7. CCS need a strategy that allows deployment of bought in solution components but retains overall control of the user experience and core data.

I

High volume markets, facilitated self service

Simple high volume low cost

Complex low volume high cost

Framework (and DPS) procurement

All need:

Contract design; direct awards; document exchange; scoring; approvals; e- auction; change log and audit; supplier and buyer directories;

Some need:

Complex interfaces, catalogues, published notices

Complex markets with CCS support

Complex markets, facilitated self service

Framework (and DPS) markets:

call-off and consumption

CCS managed framework procurement

CCS hosted framework procurement

Fig 2 - range of contract design and marketing problems

From Fig 2 it is clear that there are a range of needs. Arguably services can be delivered by different programmes as long as they are coordinated through an integrated approach. This specification should allow the different areas to work cooperatively.

## 

## Baseline

1. We have systems in the journey (Fig 3).
2. They don’t integrate well - most integrations are manual re-keying.
3. All are subject to change this year.

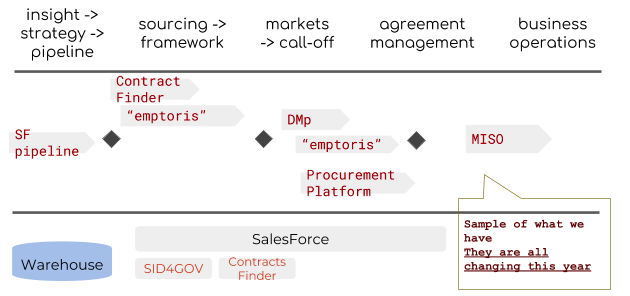


Fig 3 - sample of existing systems in context of agreement journey

# Principles

***one place to go -***

consistent design standards and look and feel

one dictionary of terms (to the user)

key once

***consistent but diverse processing -***

there *may* be more than one technical solution providing the user journey

- different user groups

- different procurement procedures e.g. OJEU procedures, call-offs

- differing levels of user procurement sophistication

- differing levels of contract complexity

- differing volumes

- differing unit value

- differing categories - e.g. travel, consulting, facilities management

journeys and state models may vary within a common core

business process solutions may need to subscribe to changes from other implementations

business process can store data but must use a common API[[2]](#footnote-2) to save changes and treat the API as the master data

***single source of truth -***

use a common API for all common data

audit all relevant data, not just the common parts

a common dictionary of coding categories must be used

a common *core* state model is required

# Service map (context)

This strategy models agreement processes in a rough journey. There may be different processes and supporting software for the same ‘part of the journey at a gross level, differing for the reasons given above. But all the processes should be supported by common data and services. Journey steps are generalised. *All terms used are pending a common agreed dictionary of terms.*

The are a number of processes in the journey of an agreement - sandwiched between a consistent user experience and common records (Fig 4.a).

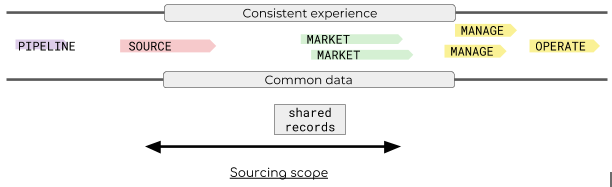


Fig 4.a - ‘sandwich’ strategy

This strategy allows there to be a range of process based solutions operating together against a common experience and with common data.

For example we may want more than one ‘market’ component - one to allow simple guided call-off procurements for high volume low cost items, and a different one for low volume high complexity call-offs.

In more detail below fig 4.b.

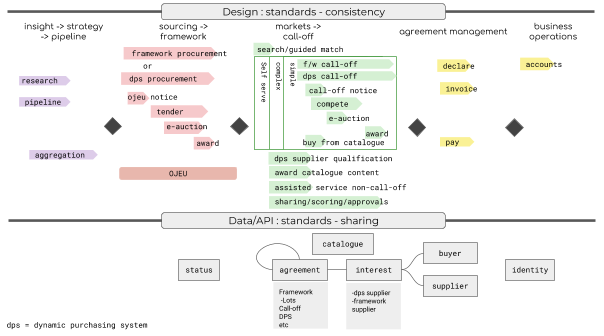


Fig 4.b - context: processes for service map including solution

approach to shared elements

Loosely, the journey is:

1. Collect intelligence and decide on framework pipeline and what to do next
2. Design, notify, tender and award a framework (including large contracts and dynamic purchasing frameworks)
3. Market the frameworks, and support call-offs, including various further competition methods, then award
4. Support the agreements which have been made/called-off, recording usage and other details, including specifically consumption to manage and invoice levy
5. Manage the organisation based on demand

This outline specification focuses on the steps **2** and the call-off elements of **3** (and facilitates 4).

Note once more that the call-off processes have varying user needs by levels of complexity and to some extent the categories of the commercial products ‘going through’ them.

# Technical Specification

The solution approach is to deliver a number of agreement ‘processing components’ to match one or several service processes from the map, and operate them in a ‘sandwich’ between Internet facing components which provide consistency in design, and record components which maintain control of core data (Fig 5.a).

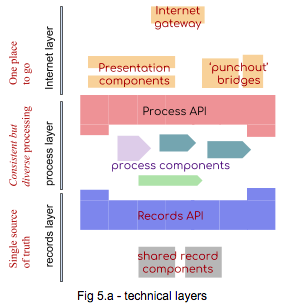
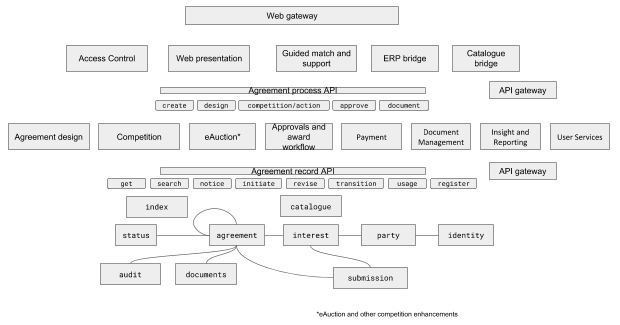


Fig 5.a - technical layers

APIs are used at each layer to standardise and control interaction. The specification is layered in a similar ‘sandwich’ way to the service map, but not drawn sequential left-to-right (Fig 5.b).

 Fig 5.b - component model of solution specification

## Description

### Internet layer

The internet layer

**Web gateway** - including firewalls, content delivery network, etc.

**Web presentation** - number of domains and subdomains with common experience, including front end of process components where required

**Access Control** - single access control for all users, possibly federated

**Guided match and support** - helps the user find the right product / process, driven by consistent product coding and keywords

**ERP bridge** - optional gateway and transformation in/out for contract initiation and consumption info

**Catalogue bridge** - optional catalogue integration with suppliers, including web and data flow

**API gateway -** (internal and external) for all APIs, rate limiting, access control etc.

**MI & Reporting** - Reporting interfaces for agreement operations

**User services** - processes for onboarding parties and individuals

### Process layer

The process layer defines an API, and some components may also present their own web interface. There may be more than one of each component, and some solution options may offer more than one component. For example a sourcing product may offer editing, competition, auction and workflow processes for OJEU procedures, but not simple directed call-off procedures at all.

**Agreement process API** - access process components in a consistent and managed way (though many interactions will be through specific web interfaces). The api should be higher level, with the correct validations, notifications and record changes made. The api should honour the procedure applicable to the agreement type - so that it only allows valid actions.

**create -** create agreement based on template in the appropriate solution

**competition/action** - api for competition based on various procedures. For example: act on an invitation to tender.

**competition/request** - create a request if in correct status and allowed by procedure - for example request for information

**competition/change** - change a request - for example change request for information details

**competition/submit -** submit a request to approval

**competition/send -** given approval send request

**competition/reply** - reply to a request

**approve** - approve a transition during competition phase. For example, scoring workflow for DOS replies before sending replies and awarding a winner.

**transition** - ask for a status transition of an agreement - for example awarding a contract.

**document** - document exchange in support of phases

**Agreement design** - create and update agreements

**Competition** - manage competition based progression of agreement status and interactions appropriate to the agreements procedure - e.g. OJEU Open Competition, FM DPS, DOS call-off, gCloud market RFI etc.

**eAuction\*** - run an e-Auction on an agreement as part of an agreement procedure. \*There may be other kinds of competition enhancements as well - such as review sites, market analysis.

**Approvals and award workflow** - given a competition state, agree on approvals (or otherwise), and status changes including award

**Document management** - allow document interaction in context of an agreement

**payment** - handle payment

**insight and reporting** - processes for insight and reporting data access and exchange

**User services** - processes for registering and helping users and organisations

### Record layer

agreement record API provides a common way to connect to and access agreement processing components.

**Agreement record API** - single API (via gateway) to update agreement data

**search-** search documents and index

**get** - get any data subject

**notice** - provide or subscribe to agreement transition notices

**register** - register a buyer or supplier

**initiate** - create a new agreement

**revise** - update any agreement without changing status subject to rights

**transition** - transition an agreement, update interests

**usage** - record agreement usage including payment, report to usage, interest, payment

*Data stores and services:*

**party** - register of all buyers, suppliers and other key roles on the agreement, linked to identities via contact preferences, plus qualification documents and declarations

**identity** - register of individuals and corporate identities, supporting contact and authentication

**agreement** - register of all agreements, including frameworks, framework lots, call-offs against frameworks, contracts, schedules and so on.

**catalogue** - storage of catalogue based items, indicating what offerings different suppliers have against the various agreements

**involvement** [was ‘interest’]- record party/role relationships between agreements and buyers/suppliers including: DPS supplier, framework supplier, buyer, contract/lot supplier.

**status** - a usually read-only register of the set of state models for various process journeys, including common subsets of common states

**audit** - a common register of append-only event records, plus process-specific supporting documents

**documents** - agreement support documents

**submission** - record of agreement usage including payment and other things

**index** - index of all key data for search - this may be spread across or synchronised with a number of solution components

Not shown: extracts to data marts; backup flows; infrastructure and zoning.

## Directives

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119](https://tools.ietf.org/html/rfc2119) when, and only when, they appear in all capitals, as shown here.

### Common directives

1. Each solution component MUST individually and jointly comply with Government Technology Code of Practice including Design Standards, Cloud Security Principles, Security Design Principles and Cloud Bulk Data Management guidance.

### Internet layer directives

1. All users MUST be authenticated and authorised via a common core Access Control component for web access.
2. All solutions MUST be delivered through a coherent set of user domains and linkage with consistent style guide and terminology.
3. All pages MUST be in sub-domains of the agreed gov.uk CCS service subdomain.
4. A common match service and documentation MUST be used to locate all agreement processes using a consistent set of product codes.
5. An ERP bridge MAY be used to integrate buyer and supplier ERP products to CCS services, initiating agreements, and updating usage information.
6. An Catalogue bridge MAY be used to integrate supplier catalogues to CCS services for the appropriate frameworks.
7. All APIs MUST be accessed via API Gateway with appropriate access control, throttling limits, caching and rout detection. There MAY be distinct Gateways for different zones (e.g. Record versus internet).
8. All APIs MUST be authenticated via an authorization token issued via a credential in the Access Control component, with the exception of Notice components for agreements that the Authority wishes to publish openly.
9. All APIs MUST include REST based JSON APIs described by OpenAPI 2 or OpenAPI 3 definitions. APIs MAY support XML and other interface standards in support of custom solutions but common data MUST also be copied to the common API.

### Process layer directives

1. The agreement process API SHOULD be based on common definitions. The definitions MAY be designed by the authority. Not all components need to support all definitions. Agreement processing components MAY offer their own API in addition to the common agreement process API but updates MUST be reflected correctly via the common API.
2. The agreement processing components MUST be accessed via the API, not directly, for computer access. Agreement processing components MAY present their own web interfaces though such interfaces MUST conform with all directives from the Internet layer.
3. All agreement processing components MUST refer to agreements, suppliers and buyers through the same reference IDs as defined in the record standard.
4. The agreement processing components MAY be shared across different processing needs. For example two different competition components may share an eAuction component.
5. There MAY be more than one of each processing components but only to service distinct needs such as different user groups or different category types, or because one component is being terminated and a new one introduced.
6. Agreement processing components MAY store processing and agreement data separate to the agreement record API but they MUST synchronise the relevant data in the agreement record API including all audit events. A component SHOULD subscribe to or poll the relevant record data where such records may change outside the component’s scope.

### Records layer directives

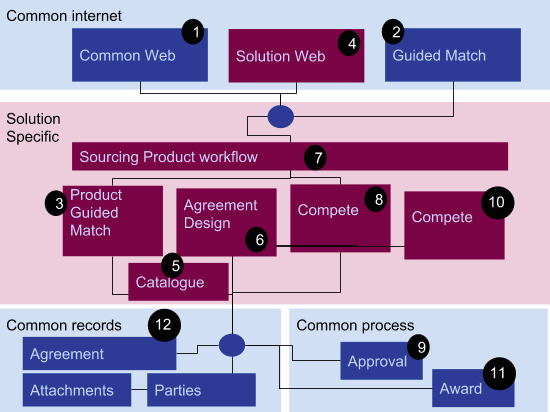
1. The agreement record API MUST be based on a common definition. The definition SHOULD be designed by the authority unless one is discovered that is already open and of sufficient quality.
2. All records MUST be version controlled.
3. Contact details SHOULD be organisation role based, not based on individual names. For example a procurement contact at supplier xyz.com should be procops@xyz.com.
4. Personal details MUST be marked as personal and MAY be redacted in indexes.
5. Commercially sensitive data MUST be marked as commercial sensitive and MAY be redacted in indexes.
6. All agreement states for an agreement MUST be referenced in a single state model document appropriate for the agreement procedure model, except for a common set of states which MUST be included from common state models. Some states MUST be included in all state models. For example an OJEU open competition agreement, when initiated, will be assigned an appropriate OJEU state model document and given the initial status from that model. The initial state could be a common initial state applied to many types of agreement and included from a common skeleton state model. This allows searching for key states across all agreements without mandating all the states in different procedures and solutions.
7. All Agreements, Buyers, Suppliers, Documents, Identities, Interests, Statuses and Agreement category codes MUST be consistently used in all entities, references and APIs.

# Solution options

1. Procure product(s) and integrate with separate environment
   1. Define or procure an environment for data and integration
   2. Procure sourcing product(s) and integrate
   3. Add / remove products as needed
2. Procure a single product that meets all needs
   1. Define standards
   2. Procure a service that fulfils standards
3. Build a product or products that meets user needs (variant of 1 or 2)

|  |  |  |
| --- | --- | --- |
|  | PROS | CONS |
| 1: Procure products and integrate with separate environment | Flexibility on products  Best of breed for different cases | Need to coordinate integration with products  Not all products will be able to integrate |
| 2: Procure a single product that meets all needs | Don’t need to integrate | Risk of lock-in |
| 3: Build a product or products that meets all needs | Exactly what we want | Rebuilding things that already exist |

Sourcing option 1 detail:



1. User starts at the front page
2. Uses global guided match with keyword or decision selection guiding towards
3. A need met by a solution specific product, which guides the user into the framework, using parameters sent from 2; further qualification questions help set up the users potential agreement need, and is
4. rendered in a subdomain using a common look and feel
5. If the framework uses a catalogue then the catalogue can provide more detail about the users need, perhaps starting a product basket
6. The guided match allows the user to start an agreement design based on the match criteria and the user completes the agreement - maybe a basket of items plus additional constraints and questions
7. The solution specific workflow can then initiate
8. the first competition phase, e.g. a Request for Proposals issue and evaluation
9. Which may need approval in a non-solution specific component
10. And potentially a second competition phase (which may be in a different solution component - e.g. an online auction
11. Which is then awarded in a common manner
12. Records are updated throughout

# Transition Approach

For option 1:

* Define the environment, standards and APIs
* Set up pipeline for integration and test of changes
* Use test environment to specify Tenders
* Ideally demonstrate integration in product selection
* Use first eSourcing tender to pilot the environment
* Add further products for CMp and other needs

Enabling steps: transition options for :

* Integrating with common identity gateway for authentication - Google Auth; SalesForce auth; SID4GOV auth
* Integration microservices backing (some) records off to SalesForce

# Evidence

### Integration approach

#### Gartner

“All too often, we hear about "us" and "them" between sourcing, procurement and vendor management teams. And teams complaining that someone in the chain "dropped the ball" or did not want to engage in a certain kind of deal. Leaders need to focus on the disciplines being an integrated capability, working like a single unit, each relying on the other for the benefit of the organization.” -- [Leadership Vision for 2018: Sourcing and Vendor Management Leader](https://www.gartner.com/document/3813765?ref=TypeAheadSearch&qid=cc5d9ccf854f198bf57b386a1)"

“Most modern ERP applications are being built (or rebuilt) using SOA and BPM concepts. “ -- [Hype Cycle for Procurement and Sourcing Solutions, 2017](https://www.gartner.com/document/3769139?ref=solrAll&refval=198945968&qid=b0b8e50c12de4c5f662d0e56bae484ef)

“Start With a Developer Portal and Work Your Way Into a Digital Marketplace by Adding More Programmable Components to APIs” -- [From APIs to Ecosystems: API Economy Best Practices for Building a Digital Platform](https://www.gartner.com/document/code/331662?ref=grbody&refval=3796563)

### Targeting solution components

Gartner

Procurement and sourcing solutions that are designed for specific categories tend to deliver more-reliable benefits because users are often quicker to realize the advantages, which helps drive adoption. -- [Hype Cycle for Procurement and Sourcing Solutions, 2017](https://www.gartner.com/document/3769139?ref=solrAll&refval=198945968&qid=b0b8e50c12de4c5f662d0e56bae484ef)

* “Traditional ERP platforms are inadequate foundations for digital business platforms.
* Properly structured, a renovated core ERP can provide the commodity business process integrity and data management needed to support digital business models.
* Ad hoc, uncoordinated ERP modernization efforts significantly hinder digital transformation efforts.”-- [Adapt Your ERP Strategy to Support a Digital Platform](https://www.gartner.com/document/3796563?ref=solrAll&refval=198949134&qid=5b6115e7be64114379e2dcff9112ad53)

### Agility

Limited time horizon before procurement PaaS arrives

Gartner / various

1. This document uses ‘agreement’ as a short term for a framework (or lot) or contract or Dynamic Purchasing solution framework [↑](#footnote-ref-1)
2. Application PRogramming Interface - a standard way of connecting systems and sending data [↑](#footnote-ref-2)