

CONSUMER  
DATA  
STANDARDS

# ENGINEERING WORKSHOP

13 June 2019

[www.consumerdatastandards.org.au](http://www.consumerdatastandards.org.au)

# WELCOME & ACKNOWLEDGEMENT

Thanking you for making the effort to be here  
and support the development of the CDS.

We acknowledge the traditional owners of the land on which  
we meet, the Gadigal people of the Eora nation and would like  
to pay our respects to their Elders past and present.

# AGENDA

Time	Activity / Talk	Presenter
14:00	Welcome	Louis Taborda
14:05	Overview of operating model	Louis Taborda
14:15	Engineering Artefacts Overview	Stuart Low
14:25	Desktop Sandbox Setup	Stuart Low
14:40	Client CLI Introduction	Stuart Low
15:10	Model Holder Design and Deployment	Stuart Low
15:20	Conformance Tools Introduction	Stuart Low
15:30	Discovery Services	Stuart Low
15:50	Client Library	Stuart Low
16:00	Code Generator for Swagger & Code	Stuart Low
16:10	Conformance	Stuart Low
16:20	Q&A	All
16:30	Conclusion	

# BACKGROUND

Data61 is supporting the development of API standards to help Australian consumers provide trusted organisations of their choice with access to their data, in safe, customer-friendly ways. Its first focus is the banking sector and subsequently to be rolled out across other designated sectors. The aim in the initial sector is to provide consumers with the ability to make informed choices about their finances: whether that's using product comparison services, getting personalised budgeting and accounting support or making decisions about the bank that's best for them

The API Standards Working Group focuses on functional requirements for open banking API's and the Information Security Standards Working Group defines the information security profile supporting the standards and authorisation and authentication flows

# PURPOSE

The purpose of this workshop is to:

- Provide the latest information and status on the Engineering work stream
- Demonstrate the desktop simulation of possible interactions between different CDR participants
- Focus on the Product APIs.



# CDS TEAM

- Program Delivery
  - Warren Bradey
  - Louis Taborda
  - Terri McLachlan
- API/ InfoSec Stream
  - James Bligh
  - Brian Kirkpatrick
- Engineering Stream
  - Stuart Low
  - Fei Yang
- Consumer Experience
  - Michael Palmyre
  - Eunice Ching

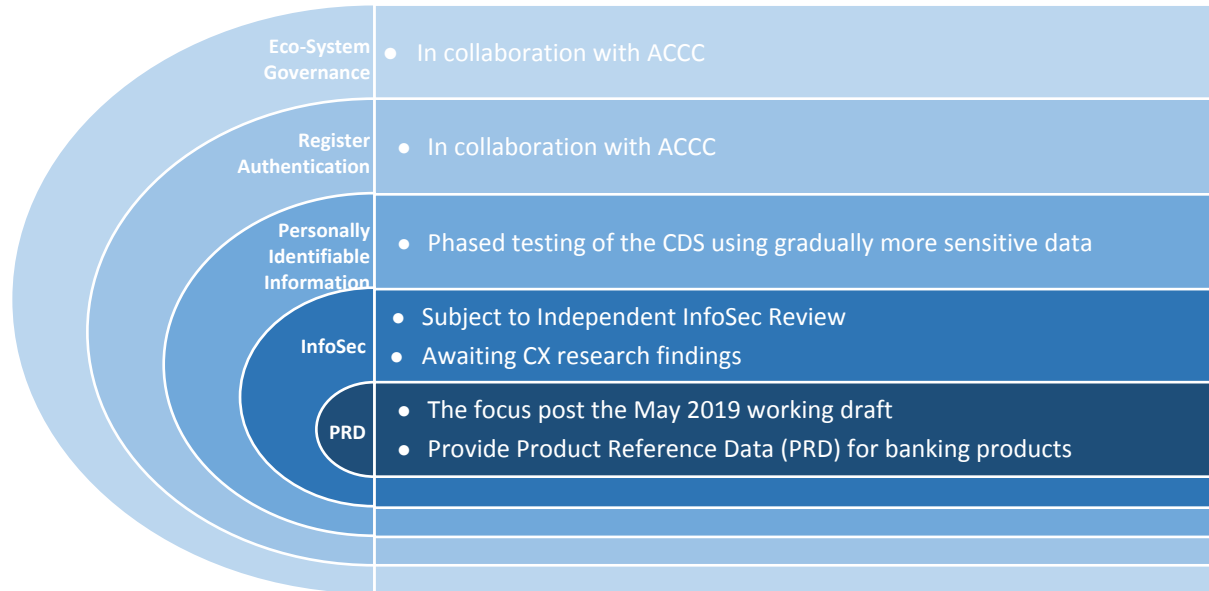


# CDS TEAM

- **Program Delivery**
  - Warren Bradey
  - Louis Taborda
  - Terri McLachlan
- **API/ InfoSec Stream**
  - James Bligh
  - Brian Kirkpatrick
- **Engineering Stream**
  - Stuart Low
  - Fei Yang
- **Consumer Experience**
  - Michael Palmyre
  - Eunice Ching

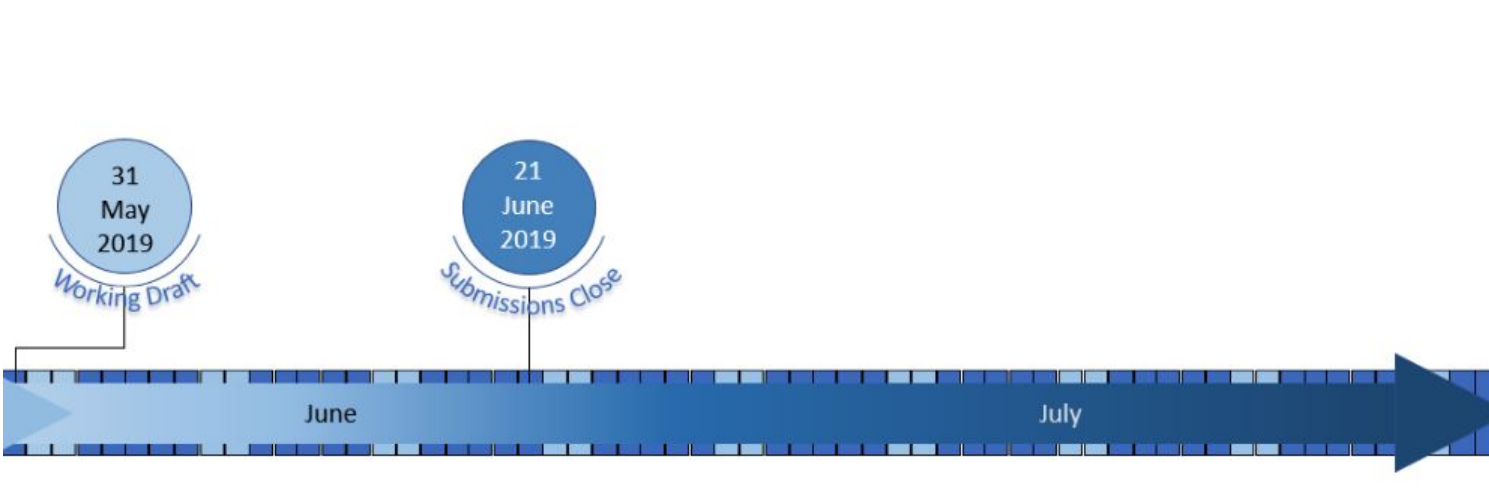


# Incremental Release/ Refinement

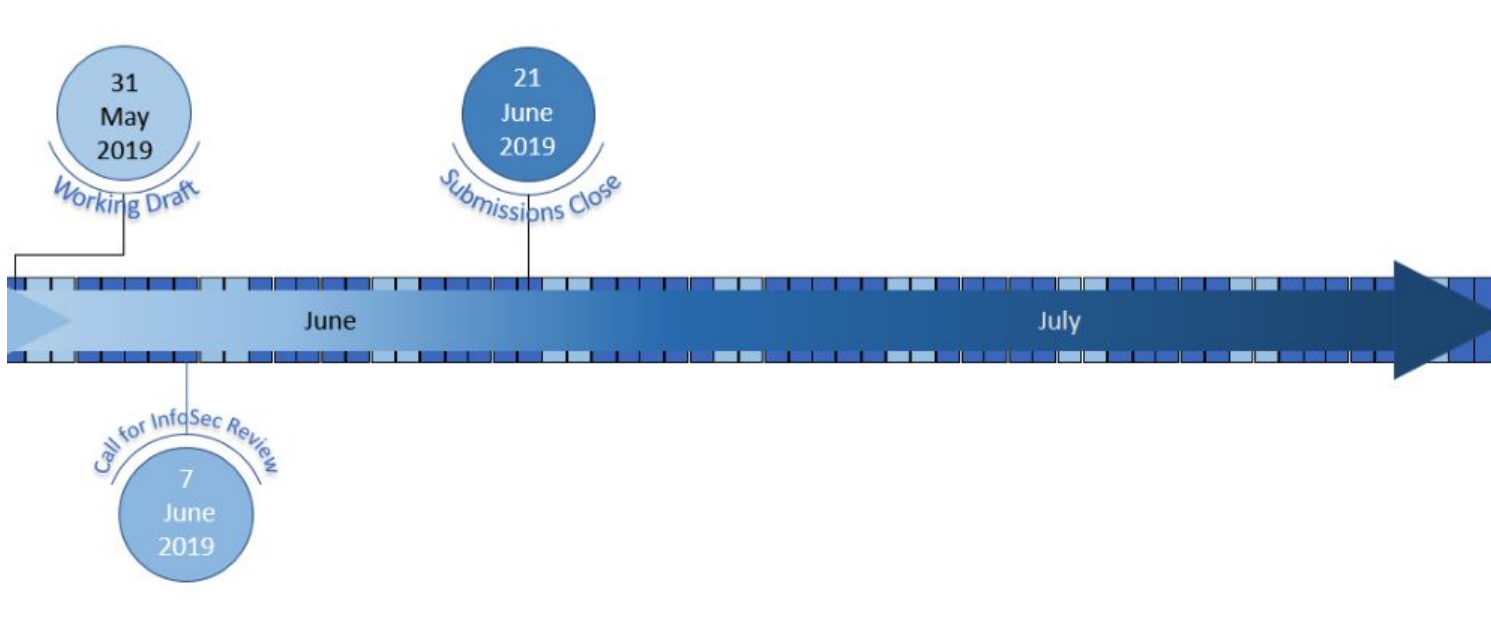




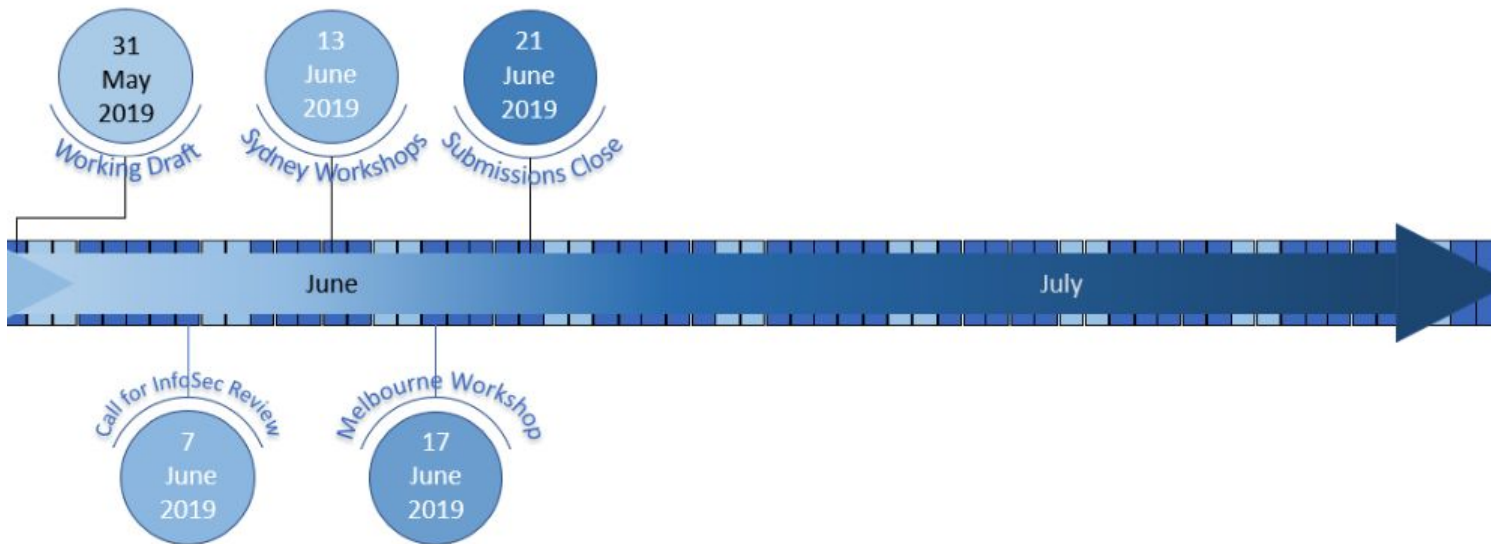
# CDS Timeline



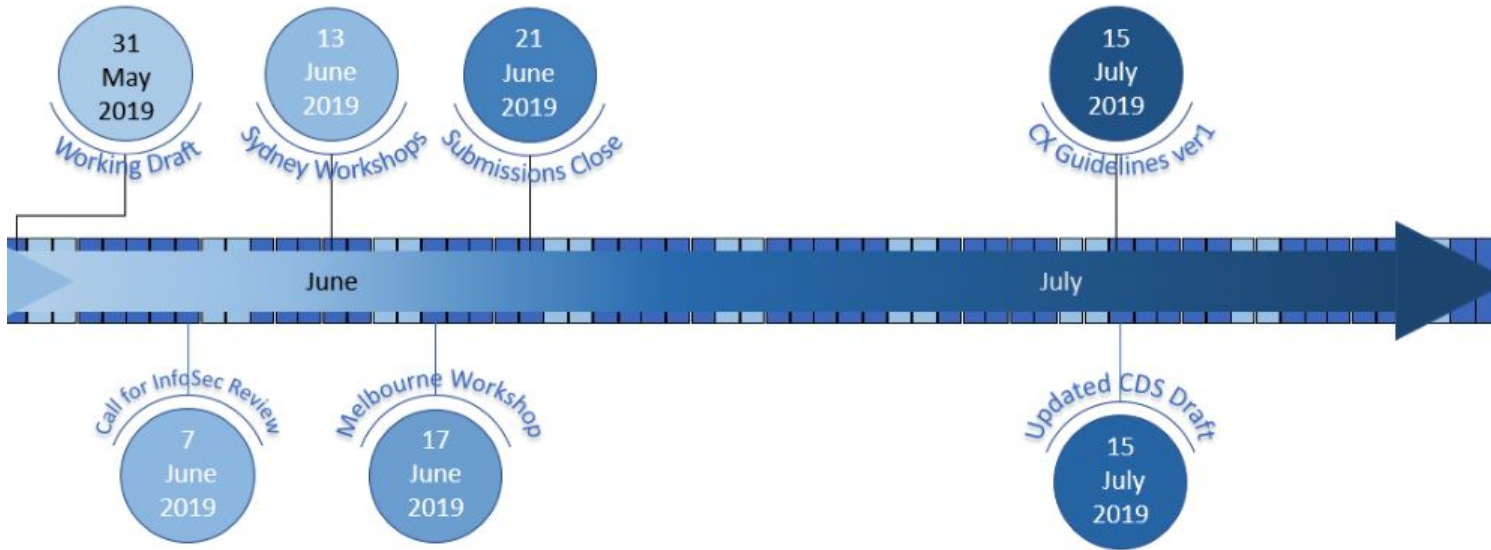
# CDS Timeline



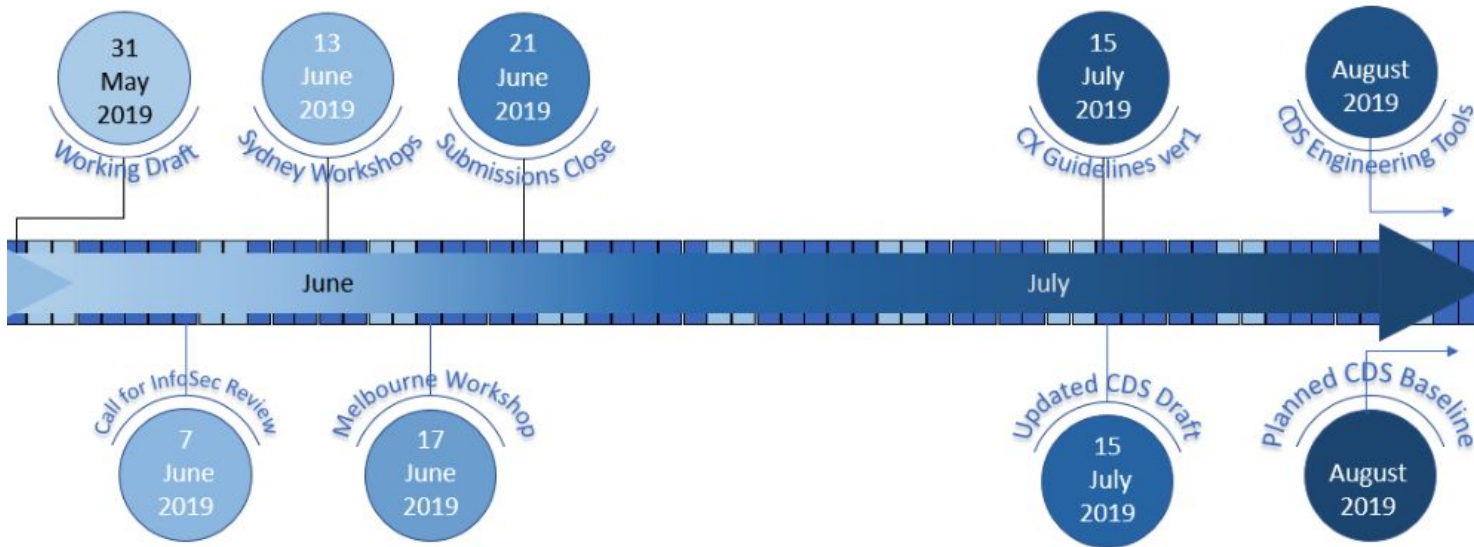
# CDS Timeline



# CDS Timeline



# CDS Timeline





Welcome to the Consumer Data Right Register GitHub

[CDR Register API](#)

[CDR Register Github](#)

[Consumer Data Standards](#)

# ACCC CDR Register

The ACCC has opened consultation on the technical design of the CDR Register.

The first round of consultation is for the CDR-Register API, which allows participants to retrieve details of Data Holder and Data Recipients. Feedback is being sought at <https://cdr-register.github.io>.

The ACCC expects to consult on other aspects of the CDR Register design, including:

- Business and technical design principles
- Security Profile and Certificate Management
- Caching and refreshing of Register metadata.

# Principles

1. Open participation
2. Transparency
3. DSB is Accountable
4. Efficient and Productive
5. Iterative and Agile



# CDS TEAM



# Operating Model – 1

CDR/CDS timelines necessitate we (all) embrace the Iterative & Agile principle:

1. Incremental releases
2. Just-in-time decision making

# Operating Model – 2

Open & Transparent will not change

CDS/ Data61 Accountability will apply as we enter “Steady State” phase with less volatility

We will be introducing:

- CDS Stream Personas
- CDS Approval Persona

Ask for your continued involvement in the CDS  
Submissions close COB 21 June 2019!

# Questions

CONSUMER  
DATA  
STANDARDS

# CDS Engineering Workshop #2

June 2019

[www.consumerdatastandards.org.au](http://www.consumerdatastandards.org.au)

# Agenda

Welcome - Louis Taborda

Engineering - Stuart Low

1. Overview
2. Desktop Sandbox Initialisation
3. Reference Client
4. Model Holder
5. Conformance Tools
6. Discovery Services
7. Extras:
  - a. Product Summariser
  - b. Server Stubs
  - c. Code Generator

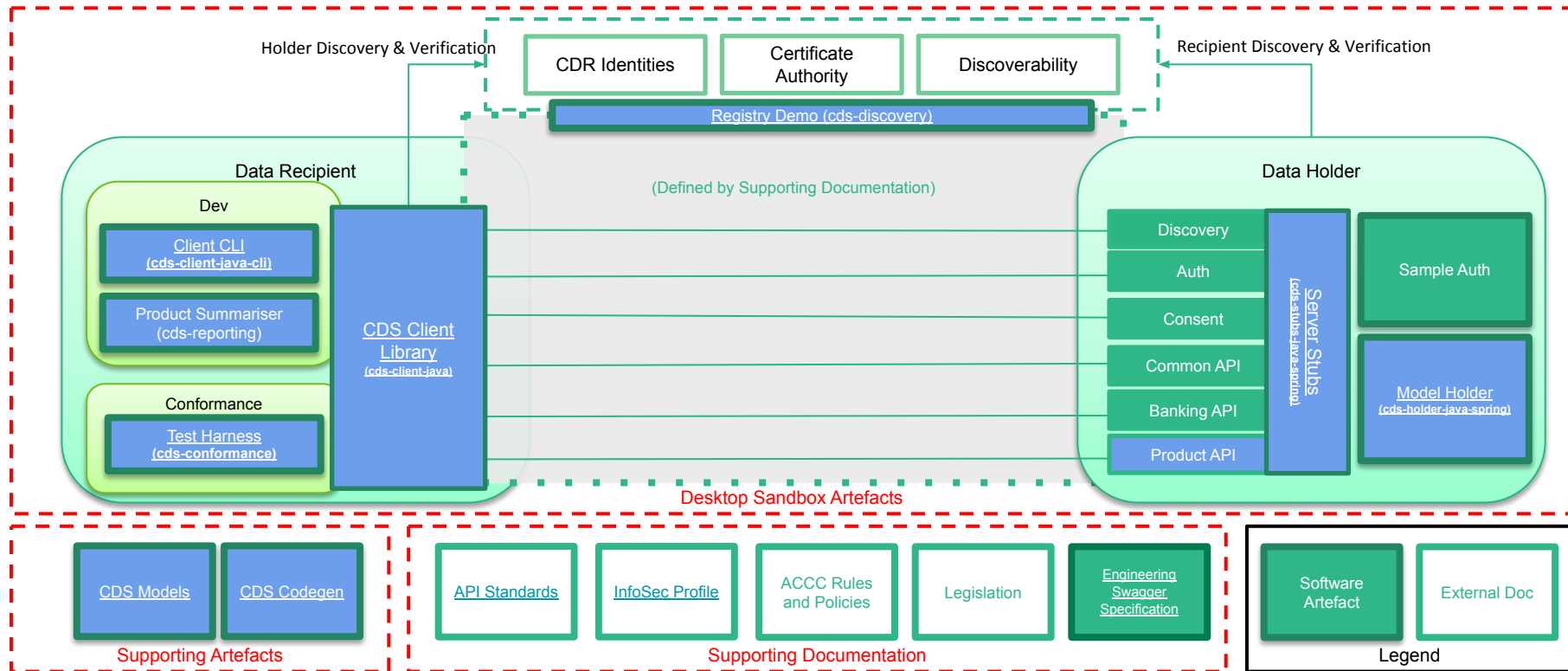
CONSUMER  
DATA  
STANDARDS

# Overview

Engineering Artefacts

[www.consumerdatastandards.org.au](http://www.consumerdatastandards.org.au)

# Artefact Map





# Artefact Overview

Quick Summary

tl;dr Getting Started

# CDS Models

A Java based representation using custom annotations for CDS specific components. Can be accessed via cds-codegen tools or directly via Java Reflections.

## Includes

- Green-tack Swagger import of 0.9.3
- Verified Product payloads between toolsets
- Definition for allOf()
- Definition for anyOf()
- Definition for conditional Business Logic Conditions and handled by cds-conformance passed through to cds-client-cli

## Does not include:

- Verified Payload Metadata for other APIs
- InfoSec Profile components (authRequired)

### Example Property using CDS Specific Validation:

```
@Property(  
    description = "Generic field containing additional information relevant  
to the depositRateType specified. Whether mandatory or not is dependent on the value  
of depositRateType",  
    requiredIf = {  
        @Condition(  
            propertyName = "depositRateType",  
            values = { "FIXED", "BONUS", "BUNDLE_BONUS",  
                    "INTRODUCTORY", "FLOATING", "MARKET_LINKED"  
        },  
        conditionalCDSDataTypes = {  
            @ConditionalCDSDataType(value = "FIXED",  
                cdsDataType = @CDSDataType(CustomDataType.Duration)),  
            @ConditionalCDSDataType(value = "INTRODUCTORY",  
                cdsDataType = @CDSDataType(CustomDataType.Duration))  
        }  
    )  
}  
)  
String additionalValue;
```

CONSUMER  
DATA  
STANDARDS

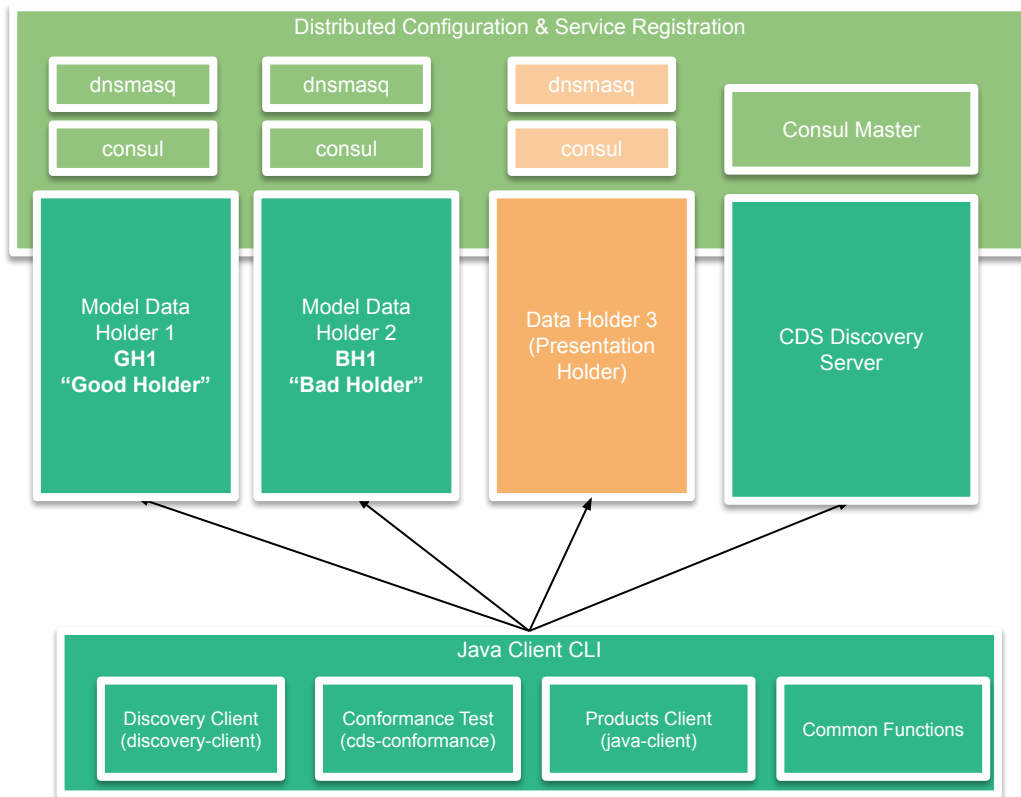
# Desktop Sandbox

Design, Architecture and Initialisation

[www.consumerdatastandards.org.au](http://www.consumerdatastandards.org.au)

# Desktop Sandbox Architecture

- Runtime started using Docker Compose (maybe kube integration works..)
- Compose includes:
  - 2 x Data Holders with:
    - Custom Payload
    - Easy add new Holder
  - Discovery Server
  - Consul store providing distributed:
    - DNS via dnsmasq
    - Configuration Store
  - Docker Network & Volume setup
- Java Client CLI can be attached seamlessly into private network
- Data mounted from [desktop-sandbox/runtime/\\*](#)



# Desktop Sandbox

Docker Compose based orchestration of the target environment for the Consumer Data Right

## Includes

- Product API for 0.9.3 Standards
- Multiple holder orchestration
- Discovery Server demonstration
- Storage persistence of consul master (ie. nodes will notice things break)

## Does not include:

- All other Standards APIs
- InfoSec Profile compliance
- Register Certificate Authority

### Clone Repository from GitHub:

```
stuart@localhost:~$ git clone
https://github.com/ConsumerDataStandardsAustralia/desktop-sandbox.git
Cloning into 'desktop-sandbox'...
<-- snip -->
Resolving deltas: 100% (241/241), done.
```

### Initiate Orchestration:

```
stuart@localhost:~$ cd desktop-sandbox
stuart@localhost:~/desktop-sandbox$ docker-compose up
Creating network "desktopsandbox_cdsdevnet" with the default driver
Creating volume "desktopsandbox_consul-data" with default driver
Creating desktopsandbox_consul_1 ...
Creating desktopsandbox_consul_1 ... done
Creating desktopsandbox_holder2_1 ...
Creating desktopsandbox_holder1_1 ...
Creating desktopsandbox_discovery_1 ...
Creating desktopsandbox_discovery_1 ... done
Creating desktopsandbox_holder2_1 ...
Creating desktopsandbox_holder1_1 ... done
Attaching to desktopsandbox_consul_1, desktopsandbox_discovery_1,
desktopsandbox_holder2_1, desktopsandbox_holder1_1
consul_1 | bootstrap = true: do not enable unless necessary
consul_1 | ==> Starting Consul agent...
consul_1 | ==> Consul agent running!
consul_1 | Version: 'v1.5.1'
consul_1 | Node ID: 'cb456d5d-9442-56f9-9c1c-df5d9bf5039a'
<-- snip -->
```

```

stuart@stuart-desktop:~/git/desktop-sandbox$ docker-compose up
Creating network "desktopsandbox_cdsdevnet" with the default driver
Creating volume "desktopsandbox_consul-data" with default driver
Creating desktopsandbox_consul_1 ... done
Creating desktopsandbox_consul_1 ... done
Creating desktopsandbox_holder2_1 ...
Creating desktopsandbox_holder1_1 ... done
Creating desktopsandbox_discovery_1 ...
Creating desktopsandbox_discovery_1 ...
Creating desktopsandbox_holder2_1 ...
Creating desktopsandbox_holder1_1 ... done
Attaching to desktopsandbox_consul_1, desktopsandbox_discovery_1, desktopsandbox_holder2_1, desktopsandbox_holder1_1
consul_1 bootstrap = true: do not enable unless necessary
consul_1 ==> Starting Consul agent...
consul_1 ==> Consul agent running!
consul_1/main/resources Version: 'v1.5.1'
consul_1 Node ID: 'cb456d5d-9442-56f9-9c1c-df5d9bf5039a'
discovery_1 ==> Starting Consul agent...
discovery_1 ==> Joining cluster...
holder2_1 ==> Starting Consul agent...
holder2_1 ==> Joining cluster...
discovery_1 Join completed. Synced with 1 initial agents
discovery_1 ==> Consul agent running!
consul_1 Node name: '1a6ce6257762'
discovery_1 Version: 'v1.5.1'
holder2_1 Join completed. Synced with 1 initial agents
discovery_1 Node ID: 'ffbedba1-5bdf-eca5-c97d-43088410646f'
discovery_1 Node name: '68a8ba8c7c48'
consul_1 Datacenter: 'dc1' (Segment: '<all>')
discovery_1 Datacenter: 'dc1' (Segment: '')
holder2_1 ==> Consul agent running!
discovery_1 Server: false (Bootstrap: false)
consul_1 Server: true (Bootstrap: true)
consul_1 Client Addr: [0.0.0.0] (HTTP: 8500, HTTPS: -1, gRPC: -1, DNS: 8600)
holder2_1 Version: 'v1.5.1'
consul_1 Cluster Addr: 10.252.252.2 (LAN: 8301, WAN: 8302) (live)
holder2_1 Node ID: '0e057bf9-ecc8-7086-8971-5160cbc88aa2'
consul_1 Encrypt: false, TLS-Outgoing: false, TLS-Incoming: false
holder2_1 Node name: '2ae66529b8ea'
discovery_1 Client Addr: [127.0.0.1] (HTTP: 8500, HTTPS: -1, gRPC: -1, DNS: 8600)

```

# Demonstration

## Desktop Sandbox Startup

docker-compose down -v && docker-compose up

# Reference Client

Client Library, Summariser, CLI and CLI Commands

# Client Library

A Reference Client written in Java for inclusion in external projects.

Generated using *cds-codegen*.

## Includes

- Products API compliant with Standards 0.9.3
  - listProducts(with, filters, and, paging)
  - getProductDetail(productId)
- Maven published artefact for inclusion

## Does not include:

- All other Standards APIs
- InfoSec Profile compliance

### Add to pom.xml dependencies:

```
<dependency>  
  <groupId>au.org.consumerdatastandards</groupId>  
  <artifactId>cds-client-java</artifactId>  
  <version>0.9.3-SNAPSHOT</version>  
</dependency>
```

### Using it:

```
ApiClient client = new ApiClient();  
client.setBasePath(holderUrl);  
BankingProductsAPI api = new BankingProductsAPI(client);  
  
ResponseBankingProductList response = api.listProducts(effective, updatedSince,  
brand, productCategory, page, pageSize);
```



```
thwart@stuart-desktop:~/git/cds-reporting$ docker run --network desktop-sandbox_cdsdevnet -v `pwd`/reporting-outputs:/opt/sphinx consumerdatastandardsaustralia/cds-product-summariser:0.9.3-SNAPSHOT -bhttp://holder1:8080/cds-au/v1 --consul.enabled=true --consulmaster=10.252.252.2

--> Starting Consul agent...
--> Joining cluster...
Join completed. Synced with 1 initial agents
--> Consul agent running!
    Version: 'v1.5.1'
    Mode ID: '962321d74fa580cf.cdba-2067f6646cd4'
    Node name: '54488664d9a'
    Datacenter: 'dc1' (Segment: '')
    Server: false (Bootstrap: false)
    Client Addr: [127.0.0.1] (HTTP: 8500, HTTPS: -1, gRPC: -1, DNS: 8600)
    Cluster Addr: 10.252.252.2 (LAN: 8300, WAN: 8300)
    Encrypt: Gossip: false, TLS-Outgoing: false, TLS-Incoming: false
    --> Log data will now stream in as it occurs:

2019/06/11 08:24:47 [INFO] serf: EventMemberJoin: 54488664d9a 10.252.252.2
2019/06/11 08:24:47 [WARN] agent/proxy: running as root, will not start managed proxies
2019/06/11 08:24:47 [INFO] agent: Started DNS server 127.0.0.1:8600 (udp)
2019/06/11 08:24:47 [INFO] agent: Started DNS server 127.0.0.1:8600 (tcp)
2019/06/11 08:24:47 [INFO] agent: Started HTTP server on 127.0.0.1:8500 (tcp)
2019/06/11 08:24:47 [INFO] agent: (LAN) joining: [10.252.252.2]
2019/06/11 08:24:47 [INFO] serf: EventMemberJoin: 9b5282029f62 10.252.252.5
2019/06/11 08:24:47 [INFO] serf: EventMemberJoin: c13305c0865 10.252.252.4
2019/06/11 08:24:47 [INFO] serf: EventMemberJoin: c6546c6066 10.252.252.3
2019/06/11 08:24:47 [INFO] serf: EventMemberJoin: 9554e4d9f09 10.252.252.2
2019/06/11 08:24:47 [INFO] agent: (LAN) joined: 1 Err: nil
2019/06/11 08:24:47 [INFO] consul: adding server 5b54a5dfc0b (Addr: tcp/10.252.252.2:8300) (DC: dc1)
2019/06/11 08:24:47 [INFO] agent: started state syncer
2019/06/11 08:24:47 [INFO] agent: Synced node info

agent:
  check_monitors = 0
  check_ttl = 0
  checks = 0
  services = 0

build:
  prerelease =
  revision = 48cc984
  version = 1.5.1

consul:
  acl = disabled
  known_servers = 1
  server = false

runtime:
  arch = amd64
  gc_count = 4
  goroutines = 47
  http_proxy = 4
  os = linux
  version = go1.12.1

serf: LAN
  coordinate_resets = 0
  encrypted = false
  event_queue = 0
  event_time = 2
  failed = 0
  health_score = 0
  intent_queue = 0
  left = 0
  member_time = 8
  members = 5
  query_queue = 0
  query_time = 1

Finished: An initial directory structure has been created.

You should now populate your master file /opt/sphinx/source/index.rst and create other documentation
source files. Use the Makefile to build the docs, like so:
  make builder
where "builder" is one of the supported builders, e.g. html, latex or linkcheck.

2019-06-11 08:24:54.185 INFO --- [ main] a.e.c.reporting.product.summary.AgUtil1 : Server URL is set to http://holder1:8080/cds-au/v1
2019-06-11 08:24:54.113 INFO --- [ main] a.e.c.reporting.product.summary.AgUtil1 : Server URL is set to http://holder1:8080/cds-au/v1
2019-06-11 08:24:55.125 INFO --- [ main] a.e.c.reporting.product.summary.AgUtil1 : Server URL is set to http://holder1:8080/cds-au/v1
```

# Demonstration

## Product Summary in 150 lines + DevOps Wrapper

docker run --network desktop-sandbox\_cdsdevnet -v `pwd`/reporting-outputs:/opt/sphinx consumerdatastandardsaustralia/cds-product-summariser:0.9.3-SNAPSHOT -bhttp://holder1:8081/cds-au/v1  
--consul.enabled=true --consulmaster=10.252.252.2  
open reporting-outputs/build/html/index.html  
open reporting-outputs/build/latex/cdsproductreport.pdf

# Client CLI

A Command Line Interface combining Client, Discovery and Conformance capabilities. Based on Spring shell which delivers tab completion and other “niceties”.

## Includes

- Products API compliant with Standards 0.9.3
- Discovery API compliant with ACCC Register
- In-line validation during product calls
- Individual payload validation (from JSON)

## Does not include:

- All other Standards APIs
- InfoSec Profile compliance

### Use Docker to pull from Docker Hub:

```
stuart@localhost:~$ docker pull
consumerdatastandardsaustralia/cds-client-java-cli:0.9.3-SNAPSHOT
0.9.3-SNAPSHOT: Pulling from
consumerdatastandardsaustralia/cds-client-java-cli
c5e155d5a1d1: Already exists
<!-- snip -->
Status: Downloaded newer image for
consumerdatastandardsaustralia/cds-client-java-cli:0.9.3-SNAPSHOT
```

### Execute in interactive shell:

```
stuart@localhost:~$ docker run -it
consumerdatastandardsaustralia/cds-client-java-cli:0.9.3-SNAPSHOT
<!-- snip -->
2019-06-10 05:52:36.508 INFO 1 --- [          main] org.reflections.Reflections      :
Reflections took 43 ms to scan 2 urls, producing 6 keys and 187 values
2019-06-10 05:52:37.092 INFO 1 --- [          main] a.o.c.client.cli.CdsClientShell  :
Started CdsClientShell in 1.762 seconds (JVM running for 2.065)
cds-shell:> help
AVAILABLE COMMANDS.....
```

# Client CLI Command List

- Spring Shell Built-Ins:
  - *help*: Show Available Commands
  - *clear, exit, quit, history, script, stacktrace*
- Common:
  - *server --url {baseUrl}*: Set target Data Holder Endpoint URL
  - *get-log-level/set-log-level*: Get/Set Minimum Log Level to Display
- Products API:
  - *list-products*: Product API 0.9.3 List Products with Filters
  - *get-product-detail --product-id {inputId}*: Retrieve Detailed Product Information for {inputId}
- Discovery:
  - *set-register-endpoint/get-register-endpoint*: Set and Display current base url for Register queries
  - *get-holder --provider-id {pid}*: Retrieve Holder information for {pid}
  - *list-holders*: List all Holders in Directory
  - *setup-holder --provider-id {pid}*: Retrieve Holder Information and set as Data Holder Endpoint
- Conformance:
  - *auto-validate --enable-validation true/false*: Enable/Disable automatic validation of all payloads
  - *conformance-status*: Current automatic validation setting
  - *validate-path -f {payload.json}* - Perform identification and validation on payload file

```
stuart@stuart-desktop:~$ docker run --network desktop-sandbox_cdsdevnet -it consumerdatastandardsaustralia/cds-client-java-cli:0.9.3-SNAPSHOT --consul.enabled=true --consulmaster=10.252.252.2
==> Starting Consul agent...
==> Joining cluster...
Join completed. Synced with 1 initial agents
==> Consul agent running!
+-----+
| Version: 'v1.5.1' |
+-----+
+-----+
| Node ID: 'f64c7a41-cab3-37e6-6efb-cc34cf01b13a' | Note: We are limiting the number of attendees to 3 per company |
+-----+
+-----+
| Node name: 'a528cf0f2bb1' |
+-----+
+-----+
| Datacenter: 'dc1' (Segment: '') |
+-----+
+-----+
| Bootstrap Server: false (Bootstrap: false) |
+-----+
+-----+
| Client Addr: [127.0.0.1] (HTTP: 8500, HTTPS: -1, gRPC: -1, DNS: 8600) |
+-----+
+-----+
| Cluster Addr: 10.252.252.6 (LAN: 8301, WAN: 8302) |
+-----+
+-----+
| Encrypt: Gossip: false, TLS-Outgoing: false, TLS-Incoming: false |
+-----+
+-----+
| Support: |
+-----+
==> Log data will now stream in as it occurs:
+-----+
| Time | Presenter | Activity |
+-----+
| 2019/06/10 09:07:17 [INFO] serf: EventMemberJoin: a528cf0f2bb1 10.252.252.6 | Louis Taborda | Welcome |
+-----+
| 2019/06/10 09:07:17 [WARN] agent/proxy: running as root, will not start managed proxies | Stuart Low | Engineering Artefacts Overview |
+-----+
| 2019/06/10 09:07:17 [INFO] agent: Started DNS server 127.0.0.1:8600 (udp) | Stuart Low | Desktop Sandbox Setup |
+-----+
| 2019/06/10 09:07:17 [INFO] agent: Started HTTP server on 127.0.0.1:8500 (tcp) | Stuart Low | Client CLI Introduction |
+-----+
| 2019/06/10 09:07:17 [INFO] agent: (LAN) joining: [10.252.252.2] | Stuart Low | Model Holder Design and Deployment |
+-----+
| 2019/06/10 09:07:17 [INFO] serf: EventMemberJoin: 68a8ba8c7c48 10.252.252.3 | Stuart Low | Conformance Tools Introduction |
+-----+
| 2019/06/10 09:07:17 [INFO] serf: EventMemberJoin: 2ae66529b8ea 10.252.252.4 | Stuart Low | Discovery Services |
+-----+
| 2019/06/10 09:07:17 [INFO] serf: EventMemberJoin: 22744f21d44b 10.252.252.5 | Stuart Low | Client Library |
+-----+
| 2019/06/10 09:07:17 [INFO] serf: EventMemberJoin: 1a6ce6257762 10.252.252.2 | Stuart Low | Client Library |
+-----+
| 2019/06/10 09:07:17 [INFO] agent: (LAN) joined: 1 Err: <nil> | Stuart Low | Client Library |
+-----+
| 2019/06/10 09:07:17 [INFO] agent: started state syncer | Stuart Low | Client Library |
+-----+
| 2019/06/10 09:07:17 [WARN] manager: No servers available | Stuart Low | Client Library |
+-----+
| 2019/06/10 09:07:17 [ERR] agent: failed to sync remote state: No known Consul servers | Stuart Low | Client Library |
+-----+
| 2019/06/10 09:07:17 [INFO] consul: adding server 1a6ce6257762 (Addr: tcp/10.252.252.2:8300) (DC: dc1) | Stuart Low | Client Library |
+-----+
| 2019/06/10 09:07:18 [INFO] agent: Synced node info | Stuart Low | Client Library |
+-----+
agent: (dc1)
+-----+
| check_monitors = 0 s |
+-----+
| check_ttls = 0 s |
+-----+
| checks = 0 |
+-----+
| services = 0 |
+-----+
build: rget
+-----+
| app: prerelease = |
+-----+
| revision = 40cec984 |
+-----+
| version = 1.5.1 |
+-----+
consul:
+-----+
| acl = disabled |
+-----+
| known_servers = 1 |
+-----+
| server = false |
+-----+
runtime:
+-----+
| arch = amd64 |
+-----+
| cpu_count = 4 |
+-----+
| goroutines = 47 |
+-----+
| max_procs = 4 |
+-----+
| os = linux |
+-----+
+-----+
| Problems |
+-----+
| 0 errors, 361 warnings, 0 others (Filter matched 100 of 361 items) |
+-----+
| Description |
+-----+
| Warnings (100 of 361 items) |
+-----+
| Attribute (frameborder) is obsolete. Its use is discouraged in HTML5 documents. |
+-----+
| Invalid location of tag (style). |
+-----+
| No end tag (s/div). |
+-----+
| No end tag (s/div). |
+-----+
Resource Path
20190505_wide /engineering/build/html/...
architecture.html /engineering/build/html/...
layout.html /engineering/source/...
layout.html /engineering/source/...
```

# Demonstration

## Docker Client CLI

docker run --network desktop-sandbox\_cdsdevnet -it consumerdatastandardsaustralia/cds-client-java-cli:0.9.3-SNAPSHOT --consul.enabled=true --consulmaster=10.252.252.2  
set-register-endpoint --discovery-endpoint http://discovery:9000/cds-register/v1

# Model Holder

Initialisation and Payload Loading

# Model Holder

A presentation focused example of the Standards from a Data Holder perspective

## Includes

- Spring Framework powered, 0.9.3 compliant Products API
- Payload Loading from Source JSONs
- Optional Service Discovery Registration via Discovery Server
- Validated Meta/MetaPaginated & Links/LinksPaginated

## Does not include:

- All other Standards APIs
- InfoSec Profile compliance
- Payload validation during loader (deliberately)
- Custom SSL Support (Yet!)

### Use Docker to pull from Docker Hub:

```
stuart@localhost:~$ docker pull
consumerdatastandardsaustralia/cds-holder-java:0.9.3-SNAPSHOT
0.9.3-SNAPSHOT: Pulling from consumerdatastandardsaustralia/cds-holder-java
c5e155d5a1d1: Already exists
<!-- snip -->
Status: Downloaded newer image for
consumerdatastandardsaustralia/cds-holder-java:0.9.3-SNAPSHOT
```

### Start service bound to port 8080 loading from ./custom-payloads:

```
stuart@localhost:~$ docker run -p 8080:8080 -v "/custom-payloads:/opt/payloads"
consumerdatastandardsaustralia/cds-holder-java:0.9.3-SNAPSHOT
<!-- snip -->
2019-06-10 06:01:42.674 INFO 7 --- [           main] a.o.c.holder.util.HolderRegistrator
: Consul is disabled, not performing service registration
2019-06-10 06:01:42.725 INFO 7 --- [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with
context path ""
2019-06-10 06:01:42.728 INFO 7 --- [           main] a.o.c.holder.HolderApplication
: Started HolderApplication in 5.536 seconds (JVM running for 5.836)
```

```

2019/06/11 08:49:34 [INFO] agent: Synced service "holder3"
2019-06-11 08:49:34.034 http://holder3:8080 INFO 45 --- [main] a.o.c.holder.util.HolderRegistrar : Holder Service registered with Consul server with id of holder3 on port 8080 with type HOLDER and service uri of http://holder3:8080
2019-06-11 08:49:34.082 INFO 45 --- [main] a.o.c.holder.util.HolderRegistrar : Holder Service Discovery payload registered as key value pair with Consul server
2019-06-11 08:49:34.123 INFO 45 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2019-06-11 08:49:34.125 INFO 45 --- [main] a.o.c.holder.HolderApplication : Started HolderApplication in 7.126 seconds (JVM running for 7.45)
2019-06-11 08:49:43.984 INFO 45 --- [0.0-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2019-06-11 08:49:43.984 INFO 45 --- [0.0-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2019-06-11 08:49:43.995 INFO 45 --- [0.0-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 10 ms
2019/06/11 08:49:44 [INFO] agent: Synced check "service:holder3"
2019/06/11 08:49:54 [INFO] serf: EventMemberJoin: 84ac73887901 10.252.252.7
2019-06-11 08:50:23.394 INFO 45 --- [0.0-8080-exec-9] a.o.c.h.a.BankingProductsApiController : Initiating product list call with supplied input of effective from CURRENT, updated since null, brand of null, product category of null for page 1 with page size of 25
2019-06-11 08:50:23.437 INFO 45 --- [0.0-8080-exec-9] o.h.h.i.QueryTranslatorFactoryInitiator : HHM000397: Using ASTQueryTranslatorFactory
2019-06-11 08:50:23.538 INFO 45 --- [0.0-8080-exec-9] a.o.c.h.a.BankingProductsApiController : Returning basic product listing page 1 of 1 (page size of 25) using filters of effective CURRENT, updated since null, brand null, product category of null
2019-06-11 08:50:23.538 INFO 45 --- [0.0-8080-exec-9] a.o.c.h.a.BankingProductsApiController : Products Page data set to isFirst: true, isLast: true
2019-06-11 08:50:29.652 INFO 45 --- [0.0-8080-exec-3] a.o.c.h.a.BankingProductsApiController : Retrieving Detailed Product Information for 51114
2019-06-11 08:50:29.671 INFO 45 --- [0.0-8080-exec-3] a.o.c.h.a.BankingProductsApiController : Found product id of 51114 and returning formatted response
2019-06-11 08:50:29.672 INFO 45 --- [0.0-8080-exec-3] a.o.c.h.a.BankingProductsApiController : Retrieved Product id 51114 with effectiveFrom: 2018-01-12T15:43:00.121Z, effectiveTo: 2020-04-12T15:43:00.121Z, lastUpdated: 2018-12-12T15:43:00.121Z
2019-06-11 08:50:34.272 INFO 45 --- [0.0-8080-exec-5] a.o.c.h.a.BankingProductsApiController : Initiating product list call with supplied input of effective from CURRENT, updated since null, brand of null, product category of null for page 1 with page size of 25
2019-06-11 08:50:34.279 INFO 45 --- [0.0-8080-exec-5] a.o.c.h.a.BankingProductsApiController : Returning basic product listing page 1 of 1 (page size of 25) using filters of effective CURRENT, updated since null, brand null, product category of null
2019-06-11 08:50:34.279 INFO 45 --- [0.0-8080-exec-5] a.o.c.h.a.BankingProductsApiController : Products Page data set to isFirst: true, isLast: true

```

# Demonstration

## Local Model Holder with Custom Payload and Register Discovery

```

docker run -h holder3 -p 8083:8080 --network desktop-sandbox_cdsdevnet -v `pwd`/runtime/holder3:/opt/payloads -it consumerdatastandardsaustralia/cds-holder-java:0.9.3-SNAPSHOT --consul.enabled=true
--consulmaster=10.252.252.2 --id=holder3 --registry.providerId=NH1

```

# Conformance Tools

Execution and Report Browsing



```

[cds-shell:]
[cds-shell:]>auto-validate false
2019-06-12 14:04:37.213 INFO 47 --- [main] a.o.c.client.cli.Conformance : Automatic payload validation setting: false
[cds-shell:]>setup-holder BH1
2019-06-12 14:04:41.839 DEBUG 47 --- [main] a.o.c.client.cli.Discovery : Identified Holder matching Provider ID of BH1 with Server URL of: http://holder2:8082/cds-au/v1
2019-06-12 14:04:41.839 INFO 47 --- [main] a.o.c.client.cli.Discovery : Discovered matching company name of: ACME Bank Pty Ltd
2019-06-12 14:04:41.839 INFO 47 --- [main] a.o.c.client.cli.Discovery : Holder Base URL identified as: http://holder2:8082/cds-au/v1
[cds-shell:]>list-products --summary
2019-06-12 14:04:45.083 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : List Products CLI initiated with effective: null, updated-since: null, brand: null, product-category: null
, page: null, page-size: null
2019-06-12 14:04:45.085 INFO 47 --- [main] a.o.c.client.cli.support.ApiUtil : Server Base URL is set to http://holder2:8082/cds-au/v1
2019-06-12 14:04:45.106 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : +-----+
2019-06-12 14:04:45.106 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : | ID | Name | Product Category |
2019-06-12 14:04:45.106 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : +-----+
2019-06-12 14:04:45.106 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : | 53645 | Acme Platinum Card | CRED_AND_CHRG_CARDS |
2019-06-12 14:04:45.107 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : +-----+
2019-06-12 14:04:45.107 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : | 52645 | Term Deposit | TERM_DEPOSITS |
2019-06-12 14:04:45.107 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : +-----+
2019-06-12 14:04:45.107 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : | 51114 | Another Every Day Account | TRANS_AND_SAVINGS_ACCOUNTS |
2019-06-12 14:04:45.107 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : +-----+
[cds-shell:]>get-product-detail 52645
2019-06-12 14:04:51.052 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : Get Product Detail CLI initiated with product id 52645
2019-06-12 14:04:51.056 INFO 47 --- [main] a.o.c.client.cli.support.ApiUtil : Server Base URL is set to http://holder2:8082/cds-au/v1
2019-06-12 14:04:51.118 INFO 47 --- [main] a.o.c.client.cli.BankingProducts : class class au.org.consumerdatastandards.client.model.ResponseBankingProductById {
data: class class au.org.consumerdatastandards.client.model.BankingProductDetail {
productID: 52645
effectiveFrom: 2018-01-12T15:43:00.121Z
effectiveTo: 2020-04-12T15:43:00.121Z

```

# Demonstration

## Docker Client CLI with Conformance

```

auto-validate false
setup-holder BH1
list-products --summary
get-product-detail 52645
auto-validate true
get-product-detail 52645

```

# Conformance Test Harness

A test case based verification for the Standards APIs.  
Accessible within *client-java-cli* and also available as a  
Serenity BDD maven goal for *HTML report*

## Includes

- Structural Conformance for 0.9.3 Product APIs
- Conditional Conformance for anyOf/allOf
- Business Rules Conformance for additional\* attributes
- Pagination & Meta Conformance
- Serenity BDD HTML Report Generation via:
  - mvn verify
  - Docker Container with Web Server

## Does not include:

- Any other APIs
- InfoSec Profile
- Possibly many more test cases

### Clone Repository from GitHub:

```
stuart@localhost:~$ git clone  
https://github.com/ConsumerDataStandardsAustralia/cds-conformance.git  
Cloning into 'cds-stubs-java'...  
<-- snip -->  
Resolving deltas: 100% (241/241), done.
```

### Build and Execute from Source:

```
stuart@localhost:~$ mvn -DskipTests=true clean install  
stuart@localhost:~$ mvn clean verify  
stuart@localhost:~$ browse ./target/site/serenity/index.html
```

### Run using Docker Container:

```
stuart@localhost:~$ docker run -p 8000:8000  
consumerdatastandardsaustralia/cds-conformance:0.9.3-SNAPSHOT -b=<baseUrl>
```

Browse to <http://localhost:8000/>

```

18:52:44,406 |-INFO in ch.qos.logback.classic.joran.action.ConfigurationAction - End of configuration.
18:52:44,407 |-INFO in ch.qos.logback.classic.joran.JoranConfigurator@2e2ff723 - Registering current configuration as safe fallback point

2019-06-11 18:52:44.720 INFO --- [main] n.t.core.steps.ConsoleLoggingListener :

-----
Serenity

News and tutorials at http://www.serenity-bdd.info
Documentation at https://wakaleo.gitbooks.io/the-serenity-book/content/
Join the Serenity Community on Rocket Chat at https://serenity-bdd.rocket.chat
Serenity BDD Support and Training at http://serenity-bdd.info/#/trainingandsupport
Learn Serenity BDD online at http://serenity-dojo.com
-----

2019-06-11 18:52:44.882 INFO --- [main] org.reflections.Reflections : Reflections took 67 ms to scan 2 urls, producing 6 keys and 187 values
2019-06-11 18:52:44.967 INFO --- [main] n.t.core.steps.ConsoleLoggingListener : Test Suite Started: Banking products API test
2019-06-11 18:52:44.991 INFO --- [main] n.t.core.steps.ConsoleLoggingListener :

TEST STARTED

TEST STARTED: listAndGet[0]
-----
2019-06-11 18:52:44.991 INFO --- [main] net.serenitybdd.core.Serenity : TEST NUMBER: 1
2019-06-11 18:52:46.958 ERROR --- [main] n.t.core.steps.ConsoleLoggingListener :

```

# Demonstration

## Conformance Tools

docker run -p 8000:8000 consumerdatastandardsaustralia/cds-conformance:0.9.3-SNAPSHOT -b=http://`hostname`:8081/cds-au/v1  
 docker run -p 8000:8000 consumerdatastandardsaustralia/cds-conformance:0.9.3-SNAPSHOT -b=http://`hostname`:8082/cds-au/v1

CONSUMER  
DATA  
STANDARDS

# Discovery Services

Simulated Directory & Discovery Client

[www.consumerdatastandards.org.au](http://www.consumerdatastandards.org.au)

# Discovery Library

A Discovery Library written in Java for identifying Holder endpoints from the ACCC Register

## Includes

- Holder Discovery APIs to **Unverified Alpha** ACCC Spec
- Maven published artefact for inclusion

## Does not include:

- Recipient Discovery (Yet)

### Add to pom.xml dependencies:

```
<dependency>
  <groupId>au.org.consumerdatastandards</groupId>
  <artifactId>cds-discovery-client</artifactId>
  <version>0.9.3-SNAPSHOT</version>
</dependency>
```

### Using it:

```
ApiClient client = new ApiClient();
client.setBasePath(discoveryUrl);
RegisterClientAPI api = new RegisterClientAPI(client);
```

```
ResponseRegisterDataHolderList holderListResponse =
api.getRegisterDataHolders(page, pageSize, updatedSince);
```

# Discovery Server

A self contained, emulation of the target ACCC Directory that uses Consul as a distributed backing store (which Model Holder registers with)

**Not a secure server. Strictly Development Purposes.**

## Includes

- **Unverified Alpha** ACCC Register API Endpoint
- Dynamic Membership reading from local Consul

## Does not include:

- Private Certificate Authority (yet)

### Use Docker to pull from Docker Hub:

```
stuart@localhost:~$ docker pull consumerdatastandardsaustralia/cds-discovery:0.9.3-SNAPSHOT
0.9.3-SNAPSHOT: Pulling from consumerdatastandardsaustralia/cds-discovery
c5e155d5a1d1: Already exists
<!-- snip -->
Status: Downloaded newer image for consumerdatastandardsaustralia/cds-discovery:0.9.3-SNAPSHOT
```

### Start Shared Consul Server:

```
stuart@localhost:~$ docker run -d -p 8301:8301 -p 8500:8500 consul agent -server -client 0.0.0.0
-bootstrap-ui
dc63c0b61dc7726f334295663dc809226b3ab6d42c694b66f28f0a069154bac7
```

### Start Discovery Service on port 9000:

```
stuart@localhost:~$ docker run -p 9000:9000
consumerdatastandardsaustralia/cds-discovery:0.9.3-SNAPSHOT --consulmaster=`hostname`
<!-- snip -->
2019-06-10 06:36:33.014 INFO 43 --- [0.0-9000-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] :
Initializing Spring DispatcherServlet 'dispatcherServlet'
2019-06-10 06:36:33.014 INFO 43 --- [0.0-9000-exec-1] o.s.web.servlet.DispatcherServlet :
Initializing Servlet 'dispatcherServlet'
2019-06-10 06:36:33.020 INFO 43 --- [0.0-9000-exec-1] o.s.web.servlet.DispatcherServlet :
Completed initialization in 6 ms
2019/06/10 06:36:33 [INFO] agent: Synced check "service:discovery.cds-discovery-server"
```

### Start Model Holder with consul registration

```
stuart@localhost:~$ docker run -p 8080:8080 -v "/custom-payloads:/opt/payloads"
consumerdatastandardsaustralia/cds-holder-java:0.9.3-SNAPSHOT --consul.enabled=true
--consulmaster=`hostname` --id=`hostname`
```

```

main] a.o.c.client.cli.Discovery : +-----+-----+
main] a.o.c.client.cli.Discovery : | Provider ID | Endpoint Address |
main] a.o.c.client.cli.Discovery : +-----+-----+
main] a.o.c.client.cli.Discovery : |          AC1 | http://holder1:8081/cds-au/v1 |
main] a.o.c.client.cli.Discovery : +-----+-----+
main] a.o.c.client.cli.Discovery : |          AC2 | http://holder2:8082/cds-au/v1 |
main] a.o.c.client.cli.Discovery : +-----+-----+

main] a.o.c.client.cli.Discovery : Discovered matching company name of: ACME Bank Pty
main] a.o.c.client.cli.Discovery : Holder Base URL identified as: http://holder1:8081/

```

## Services 4 total

service:name tag:name status:critical search-term

Service	Type	Health Checks ⓘ	Tags
consul		✓ 1	
discovery		✓ 2	DISCOVERY
holder1		✓ 2	HOLDER
holder2		✓ 2	HOLDER

## Consumer Data Standards Discovery API®

[ Base URL: localhost:9000/cds-register/v1 ]  
<http://localhost:9000/2api/docs>

Simulation of the emergent ACCC Registry Discovery APIs

[Terms of Service](#)

[Data#1 - Website](#)

[Send email to Data#1](#)

[MIT License](#)

**register-api-controller** Register Api Controller

**Register**

Models

# Demonstration

## Discovery Services

<http://localhost:9000/swagger-ui.html>

<http://localhost:8500/ui/dc1/services>

localhost:8500/ui/dc1/kv/holder1/edit

dc1 Services Nodes **Key/Value** ACL Intentions

< Key / Values

### holder1

Value

```
{
  "accreditation": null,
  "participant": {
    "name": "ACME Bank",
    "participantId": null,
    "participantType": "HOLDER",
    "organisation":

```

# Thanks

Also to those who have been contributing, we welcome more on GitHub Engineering  
Special thanks to [@fyang1024](#)



CONSUMER  
DATA  
STANDARDS

# Extras

Stubs, Doc Generator, Codegen

[www.consumerdatastandards.org.au](http://www.consumerdatastandards.org.au)

# Product Summariser

Basic Template based Product output generator.  
Produces Sphinx reStructured Text to generate HTML and pipeline it to a PDF.

Uses cds-client-java

## Includes

- Detailed Product output in HTML
- Detailed Product output as PDF

## Does not include:

- Much else, starting point, demo of client artefacts (only 150 lines of Java).

### Run Summariser with Docker and export to reporting-outputs/

```
stuart@localhost:~$ docker run --network desktop-sandbox_cdsdevnet -v  
`pwd`/reporting-outputs:/opt/sphinx  
consumerdatastandardsaustralia/cds-product-summariser:0.9.3-SNAPSHOT  
-b=http://holder1:8081/cds-au/v1 --consul.enabled=true --consulmaster=10.252.252.2
```

### Open Result Reports:

```
stuart@localhost:~$ open reporting-outputs/build/html/index.html  
stuart@localhost:~$ open reporting-outputs/build/latex/cdsproductreport.pdf
```

# Java Server Stubs

A skeleton Spring service for Standards

Generated using *cds-codegen*.

## Includes

- Spring powered 0.9.3 compliant Products API
- Response models to populate
- Placeholders to inject business logic

## Does not include:

- Any functionality beyond Spring Framework setup & model setup
- All other Standards APIs
- InfoSec Profile compliance

### Clone Repository from GitHub:

```
stuart@localhost:~$ git clone
https://github.com/ConsumerDataStandardsAustralia/cds-stubs-java.git
Cloning into 'cds-stubs-java'...
<-- snip -->
Resolving deltas: 100% (241/241), done.
```

### Start service on port 8080:

```
stuart@localhost:~$ cd cds-stubs-java/
stuart@localhost:~/cds-stubs-java$ mvn clean spring-boot:run
<-- snip -->
2019-06-10 16:11:11.383 INFO 1408 --- [          main] s.d.s.w.s.ApiListingReferenceScanner
: Scanning for api listing references
2019-06-10 16:11:11.598 INFO 1408 --- [          main]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with
context path ""
2019-06-10 16:11:11.601 INFO 1408 --- [          main] a.o.c.stubs.HolderApplication
: Started HolderApplication in 3.639 seconds (JVM running for 6.905)
```

# Code Generator

A multi purpose CDS structure aware code generation aware tool.

## Includes

- Full CDS Models Support
- Per API Filtering
- Generator for strong & rigid swagger creation
- Code generator plugin for:
  - Used by CDS for Automatic Components
  - Velocity Template based
  - Custom Templates available

## Does not include

- Good Documentation (YET!)

### Compile from Source:

```
stuart@localhost:~$ git clone https://github.com/ConsumerDataStandardsAustralia/cds-codegen.git
stuart@localhost:~$ cd cds-codegen
stuart@localhost:~$ mvn clean package -Ddockerfile.skip=true
```

### Use Docker to run Codegen to Swagger (with BankingProducts filter):

```
stuart@localhost:~$ docker run -v /tmp/codegenoutput:/opt/output
consumerdatastandardsaustralia/cds-codegen-core:0.9.3-SNAPSHOT -i BankingProducts -o
/opt/output/swagger.json
```

### Run Codegen to Swagger (with BankingProducts filter):

```
stuart@localhost:~$ cd cds-codegen
stuart@localhost:cds-codegen stuart$ java -jar
modules/cds-codegen-core/target/cds-codegen-core-0.9.3-SNAPSHOT.jar -i "BankingProducts" -o
/tmp/swagger.json
2019-06-12 21:54:03.383 INFO --- [      main] a.o.c.c.g.openapi.SwaggerGenerator : Successfully
wrote swagger to /tmp/swagger.json
```

### Run Codegen to CDS Client Java (with BankingProducts filter):

```
stuart@localhost:~$ cd cds-codegen
stuart@localhost:cds-codegen stuart$ java -jar
modules/cds-codegen-core/target/cds-codegen-core-0.9.3-SNAPSHOT.jar -i "BankingProducts" -g
au.org.consumerdatastandards.codegen.generator.code.CodeGenerator -t cds-client-java -o
/tmp/outputgen
2019-06-12 21:58:15.151 INFO --- [      main] org.reflections.Reflections : Reflections took 79 ms
to scan 1 urls, producing 6 keys and 187 values
2019-06-12 21:58:16.402 INFO --- [      main] a.o.c.c.generator.code.VelocityHelper : Wrote file:
/tmp/outputgen/src/main/java/au/org/consumerdatastandards/client/ApiCallback.java
2019-06-12 21:58:16.529 INFO --- [      main] a.o.c.c.generator.code.VelocityHelper : Wrote file:
/tmp/outputgen/src/main/java/au/org/consumerdatastandards/client/ApiClient.java
<snip>
```