CHESS Replacement Project

Connectivity & Integration Working Group

9 October 2019

ASX

Important Information – Competition Law Policy

Working group members are reminded to have regard to their obligations under competition law. In particular, please note recent changes to the Competition and Consumer Act to prohibit a corporation from engaging with one or more persons in a concerted practice that has the purpose, effect or likely effect of substantially lessening competition.

Agenda

- CDE Update
- Testing Support CDE Release 4
- CDE Release 4 Security Uplift
- Connectivity Updates
- Extractor
- Questions

CDE Update

CDE Function Availability

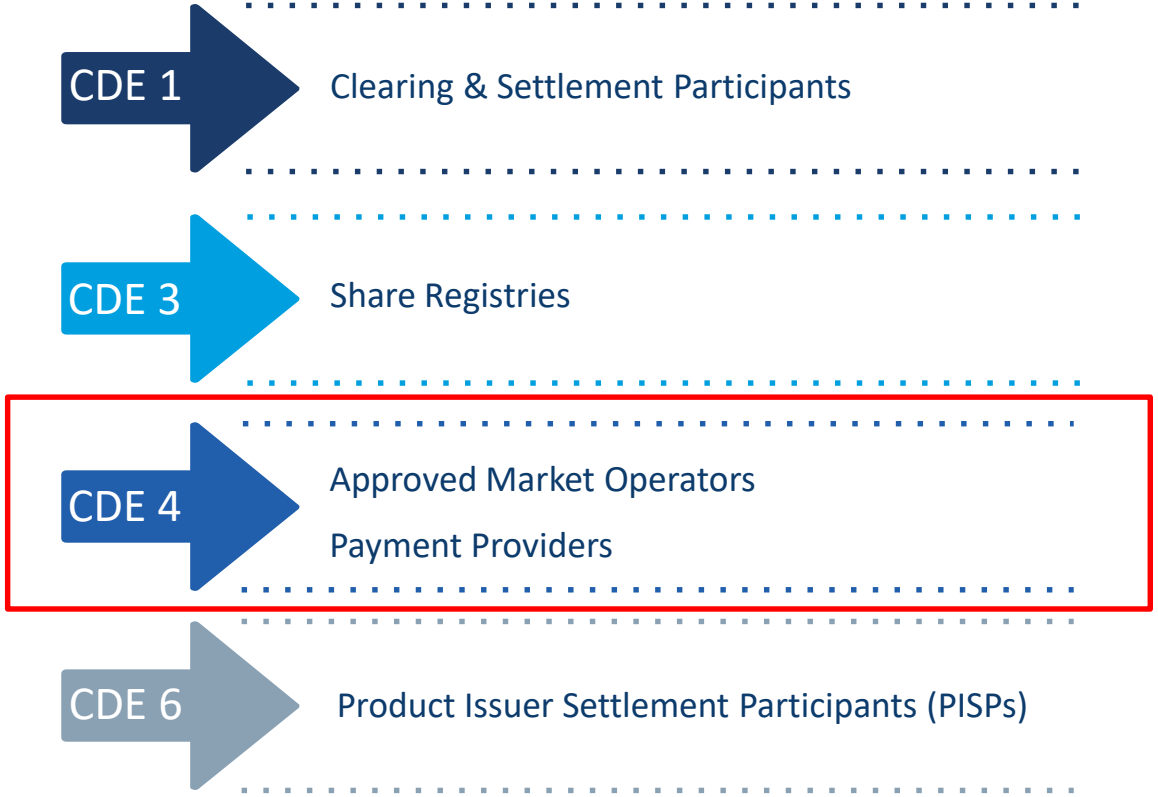
Function	Drop 1 (April'19)	Drop 2 (June'19)	Drop 3 (Aug'19)	Drop 4 (Oct'19)
Unilateral Demand Transfers	X	X	X	X
Bilateral Demand Transfers	X	X	X	X
Bilateral Demand Settlement Instructions	X	X	X	X
Batch Settlement (Payment Facilities)		X	X	X
Market Trade (Capture, Registration, Novation)		X	X	X
Account Management (Account and Holder Creation, Account and Holder Modification)			X	X
Netting & Reversal			X	X
Transfers / Conversions Between Sub-registers			X	X
SRN Enquiry			X	X
Unilateral Settlement Instruction				X
Bilateral Settlement Instruction				X
Batch Settlement (Funds Obligation, Movement of Units, Settlement Confirmation, Cancellation)				X
Corporate Actions (Ex-period, Basis of Movement, Security Sub-register Status Notification)				X

CDE Function Availability (continued)

Function	Drop 5 (Dec'19)	Drop 6 (Feb'20)	Drop 7 (May'20)
Change of Controlling Participant	X	X	X
Settlement Locks on HIN Holdings	X	X	X
Issuer's Agent Transfers (Warrants)	X	X	X
Give-ups / Take-ups		X	X
Corporates Actions (Takeovers & Buy-backs, Reconstructions, Holding Adjustments, Schemes of Arrangement)		X	X
Collateral Management (ETO & CMM)		X	X
Batch Settlement (Unit Fails)		X	X
Primary Market Facility (Placements)		X	X
Real-Time Settlement: Payment Facilities		X	X
mFunds (Foundation set up / onboarding)		X	X
Account Management (Account and Holder Cancellation, Locking / Unlocking, Investor Data and Pass Through Information)			X
Corporate Actions (Code Change, Class Merger, Diary Adjustments, DRP Elections and Election & Payment for Exercises)			X
mFunds (Applications, Redemptions and Transfers)			X
Early Settlement			X
Demand Reporting			X

Refer to - https://asxchessreplacement.atlassian.net/wiki/spaces/CSP/pages/23233327/Forward+Release+Plan

Connectivity Entry Points



Test Support CDE 4

Trade Ingest API

Features

- Will be released as a new feature for CDE release 4
- Provide participants ability to self-serve a richer data set of trade information
- Participants can submit a CSV formatted file containing trade information
  - Submit programmatically using cURL
- ASX will continue to support loading a limited subset of trades for participants as well
- Trade self-service is an opt-in service, to opt-in
  - If currently enrolled in CDE – contact CSP support and request trade ingest API service
  - If not currently enrolled in CDE – select self-serve trade option on registration form

TRADEID (1003)	SYMBOL (55)	LASTPX (31)	LASTQTY (32)	BUYPARTYIDEXEC(448)	SELLPARTYIDEXEC (448)
ATEST1123456	CBA	50.000001	1000	3614	1402
ATEST2123456	CBA	50.000001	10000	3614	1402
ATEST3123456	CBA	50.000001	1000	3614	1402
ATEST4123456	CBA	50.000001	1000	3614	1402

Sample file, not all columns displayed

Corporate Actions

Features

- CDE release 4 contains new support for corporate action ex-periods
- CDE 4 will have a number of corporate actions seeded on creation of the environment
- Stock codes with ex periods will be listed on the technical documentation website
- Ex-periods defined for dividends, bonus and rights issues
- Participants can upload trades through the Trade Ingest API with those stock codes if they wish to test ex-periods
- Ability to settle and transfer units using stocks in ex-period

Corporate Action Calendar

Day 0	Day 1	Day 2
Last "cum" quote	Ex-period	
	Ex-date	
	1st "ex" quote	
	Create Cum Entitlements.	
	CSP maintains cum entitlement balances	
	Transactions may be settled "cum" or "ex"	
	Most "cum" transactions settle	
		Record date
		CSP sends cum entitlement balance information to the registry during the ex-period until end of day on record date. The final cum entitlement balance is as at End of Day on Record Date.

Payment Providers

Features

- Batch settlement will run automatically daily from CDE release 4
- This will include settlement obligations and settlement instructions

- For Participants
  - Cash settlement with payment providers will be simulated
- For Payment Providers
  - Fund obligations instructions will simulated each settlement date and sent to Payment Providers

Current CDE Security Model

Current Security

Ledger API

- Server-side TLS – encrypted connection between customer and ASX
- Each customer connects to a unique URL cdeXXX.ledgerapi.csp.gcp.cloud.asx.com.au:3XXX
- Resolve URL locally to 203.4.179.117:3XXX
- Each customer is connected to their own CDE instance

AMQP

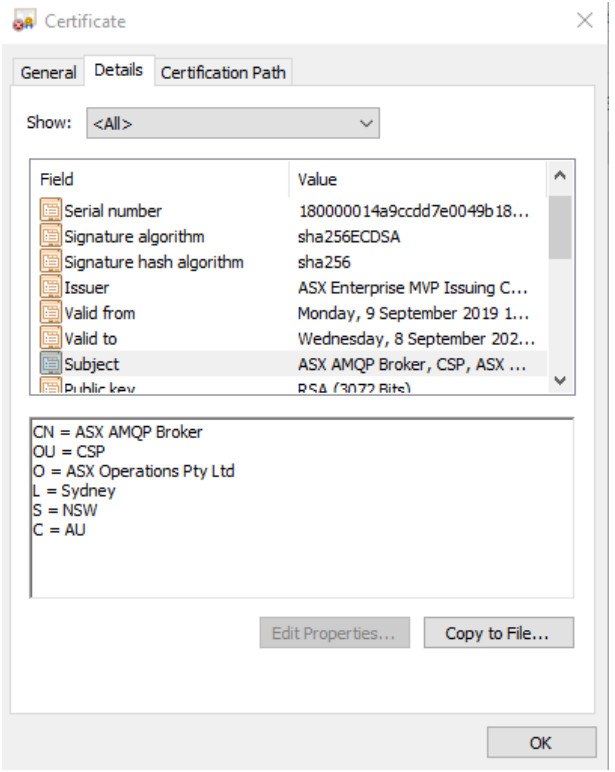
- User name and password provided at registration
- Each customer connects to same URL cde-amqp.asx.com.au:4005
- Resolve URL locally to 203.4.179.118:4005
- Each customer is authenticated onto their own request and notification queues

CDE Release 4
Security Changes

ASX TLS Certificates

TLS certificate features

- Enables transport layer encryption
- Secures server to server communication
- Consists of a private key, public key, and issuing certificate authority public key
- ASX will provide CHESS users with certificate authority public keys
- ASX enabling mutual TLS on all connectivity channels from CDE release 4, customers must update security to continue connectivity
- ASX supporting TLS versions 1.2 and 1.3
- ASX rejecting requests of TLS v1.1 and below



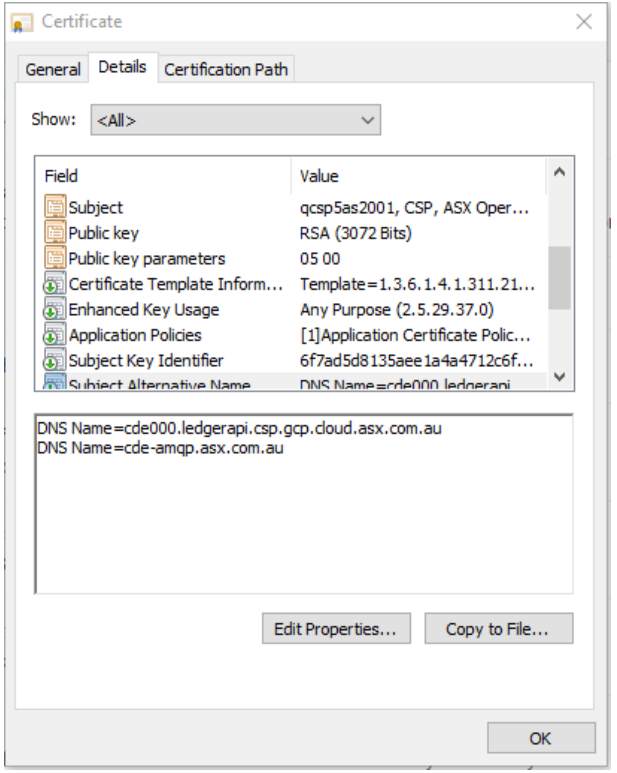
Customer TLS Certificates

TLS certificate features

- Allow participants ledger API access to a defined environment
- Participant certificates bound to ASX Net IP addresses

TLS certificate creation

- Participants
  - Generate a certificate signing request (CSR)
  - Email CSR to cts@asx.com.au
- ASX
  - Validates request
  - Signs public key
  - Return signed public key with root and issuing CA public keys
- Questions, contact CTS 02-9227-0372



CSR Configuration File

openssIConfig.txt

C: > Users > rahuba_j > Downloads > openssIConfig.txt

```
[ req ]
default_bits = 3072
prompt = no
default_md = sha256
distinguished_name = dn
req_extensions = req_ext

[ dn ]
CN=cdeXXX
OU=CSP
O=ASX Operations Pty Ltd
L=Sydney
ST=NSW
C=AU

[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = cdeXXX.ledgerapi.csp.gcp.cloud.asx.com.au
DNS.2 = cde-amqp.asx.com.au
DNS.3 = cde-trade-ingest.asx.com.au
DNS.4 = cde-fix.asx.com.au
```

OpenSSL configuration file

- Allows for reproducible CSR creation
- Receives series of default parameters which are sent to OpenSSL during key and CSR creation

Customer details

- CN = CDE instance (replace XXX with instance number)
- OU = CSP
- O = Organization linked to CDE instance (replace with your organization's name)

Extension information

- Subject alternative names – used by ASX to authorize to complete TLS handshake and authorize participant onto services
- Replace XXX with your CDE instance number

Generating a CSR

Generate new public / private key pair using configuration file

- `openssl req -out $csrOutputFileName.csr -new -nodes -keyout $privateKeyOutputFileName.key -config $configFileName -passout stdin`
- Enter password for private key

```
[dltuser@qcsp5ab2001 csr]$ ll
total 4
-rw-------. 1 dltuser dltuser 332 Oct  3 14:12 csrreq.conf
[dltuser@qcsp5ab2001 csr]$ openssl req -out testCSR.csr -new -nodes -keyout testPrivateKey.key -config csrreq.conf -passout stdin
password
Generating a 3072 bit RSA private key
......++
.....................................................................................................++
writing new private key to 'testPrivateKey.key'
-----
[dltuser@qcsp5ab2001 csr]$ ll
total 12
-rw-------. 1 dltuser dltuser  332 Oct  3 14:12 csrreq.conf
-rw-------. 1 dltuser dltuser 1496 Oct  3 14:52 testCSR.csr
-rw-------. 1 dltuser dltuser 2484 Oct  3 14:52 testPrivateKey.key
[dltuser@qcsp5ab2001 csr]$
```

Viewing CSR and Private Key

View contents of public key

- `openssl req -text -noout -verify -in $csrName.csr`

```
[dltuser@qcsp5ab2001 csr]$ openssl req -text -noout -verify -in testCSR.csr
verify OK
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: CN=cde000, OU=CSP, O=ASX, L=Sydney, ST=NSW, C=AU
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (3072 bit)
                Modulus:
                    00:ef:a5:01:0c:8e:9c:2f:33:64:df:80:02:6b:10:
                    c3:71:41:24:3a:26:fa:38:15:1f:31:da:64:c7:09:
```

```
                    d3:ec:da:b7:ec:ea:c1:7f:b7:16:b2:7b:b6:05:a2:
                    d0:83:0c:37:8a:9d:0e:b3:da:73
                Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
            X509v3 Subject Alternative Name:
                DNS:cde000.ledgerapi.csp.gcp.cloud.asx.com.au, DNS:cde-amqp.asx.com.au, DNS:cde-trade-ingest.asx.com.au
    Signature Algorithm: sha256WithRSAEncryption
         ab:1e:aa:c9:2a:d2:ae:0b:82:f1:8e:77:ab:33:fd:c1:a5:13:
         3a:5b:90:3a:35:01:bb:9d:5f:f8:2f:9f:3f:5d:58:09:dc:cb:
```

Viewing CSR and Private Key

View contents of private key

- `openssl rsa -in $privateKeyName.key -check`

```
[dltuser@qcsp5ab2001 csr]$ openssl rsa -in testPrivateKey.key -check
RSA key ok
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIIG4gIBAAKCAYEA76UBDI6cLzNk34ACaxDDcUEkOib6OBUfMdpkxwkYoyG1WowU
66lguADnckNVA9DmzQtiD+JaEwn0M9jcYsuH+NO/0ht1oqhBJxrFKG3tJh6jitNw
IAmRTtyU5lO9he0JBPXdyg8i83q+i/YU01c7XNX5fNSCgoC+X7tOa5mJ0vpH3RZC
k4wK1YkdGvgGKi6SzNUpuTGHJgip+DNlZETWIybIttz2R6vUGwEq33MGoxsLa9Nb
```

Submit CSR to ASX for validation and signing

- Email to cts@asx.com.au
- **Add ASX's root and issuing CA public keys to your local and Java trust store**
- `keytool -keystore $trustStoreName.jks -alias root -import -file $certChainFileName.cer`
- `keytool -import -alias $asxCertChainAlias -keystore /usr/java/latest/jre/lib/security/cacerts -file $certChainFileName.cer`

Helpful keytool and openssl commands

View contents of a keystore

- `keytool -list -v -keystore $keystoreName.jks`

View a certificate

- `keytool -printcert -v -file $certToView.cer`

View contents of public key in a CSR

- `openssl req -text -noout -verify -in $csrName.csr`

View contents of public key in a CSR

- `openssl req -text -noout -verify -in $csrName.csr`

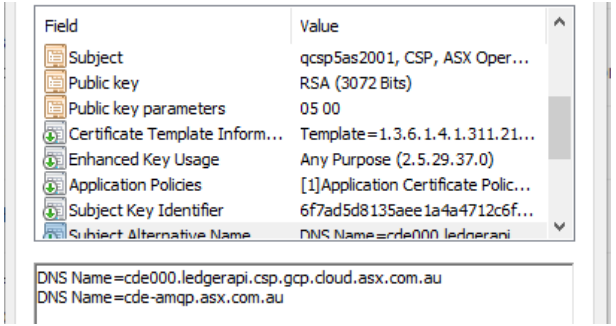
Receive public key of a server

- `openssl s_client -connect $host:$port`

Ledger API

Security

- Mutual TLS v1.3 or v1.2– server and client authenticate using TLS certificate
- TLS certificate mapped to customer's IP range
- Domain name in SAN checked against requested environment name



Channel Builder Updates

```
private ManagedChannel getChannel(String host, int port) {
    int MAX_MESSAGE_SIZE = 128 * 1024 * 1024;
    channel = NettyChannelBuilder.forAddress(params.getHost(), params.getPort())
        .sslContext(GrpcSslContexts.forClient()
            .trustManager(new File(trustStore))
            .keyManager(new File(keyStore),
                new File(privateKey))
        .keepAliveTimeout(100, TimeUnit.SECONDS)
        .keepAliveWithoutCalls(true)
        .maxInboundMessageSize(MAX_MESSAGE_SIZE)
        .build());
    return channel;
}
```

AMQP

Security

- Mutual TLS v1.2 – server and client authenticate using TLS certificate
- TLS certificate mapped to customer's IP range
- TLS certificate belongs to an AMQP user, each user has a role, and each role has queue permissions
- Removes need for user name and password

AMQP URL Updates

```
amqps://cde-amqp.asx.com.au:4005?
    sslEnabled=true                          #indicates to broker to use TLS
    &trustStorePath=/trust/store/path        #location of client trust store with issuing CA public key
    &trustStorePassword=$trustStorePassword  #recommend using a configuration and not actual password
    &keyStorePath=/key/store/path            #location of client key store with private and public key
    &keyStorePassword=$keyStorePassword      #recommend using a configuration and not actual password
    &saslMechanisms=EXTERNAL                 #indicates to broker to use client TLS certificate for authentication
```

Trade Ingest API

Security

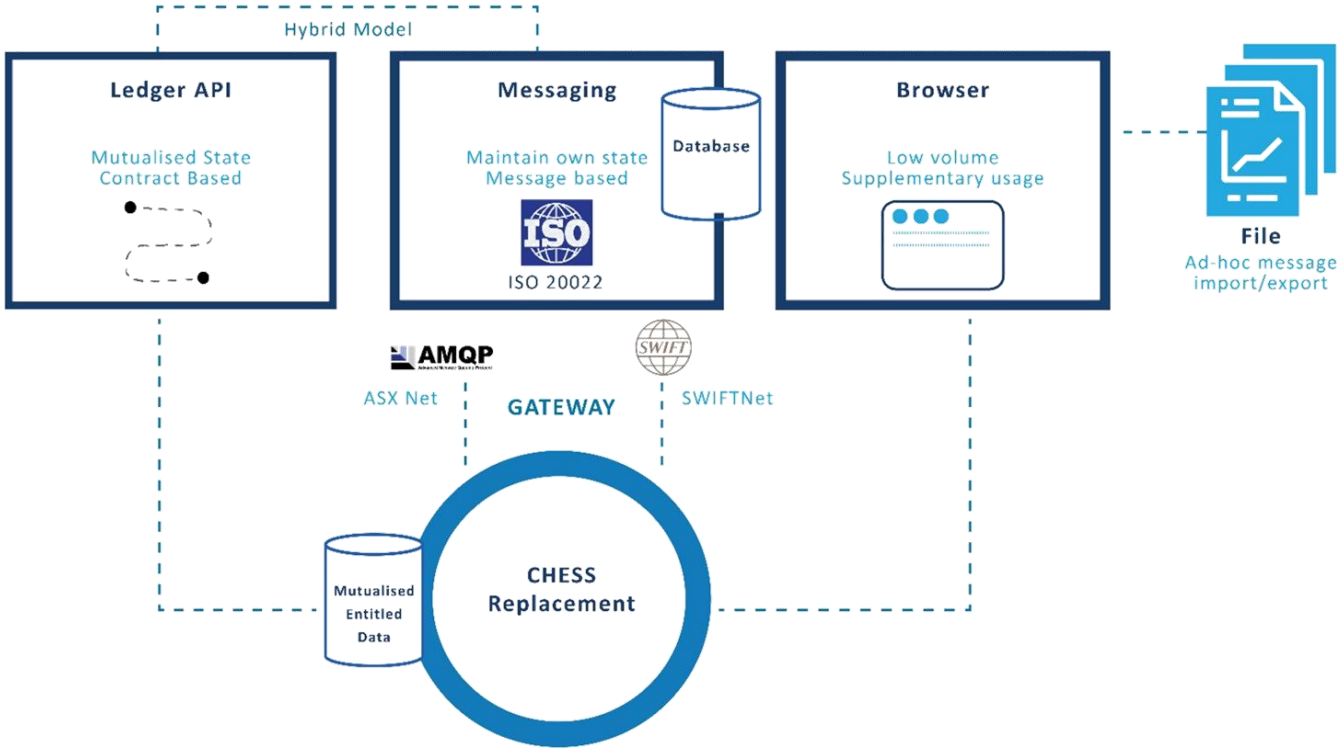
- Mutual TLS v1.2 – server and client authenticate using TLS certificate
- TLS certificate mapped to customer's IP range
- Each customer can submit a csv file of trades to their own CDE instance

cURL Connectivity

```
curl \
     --cert $clientCert:$certPassword \          #file containing customer public and private key
     --capath /path/to/CA/file \                 #location containing ASX's root CA's public keys
     --cacert $caCertToUse.pem \                 #ASX's CA's public keys in pem format
     -F 'csvFile=@/path/to/csv/file' \           #location of CSV file to upload
     https://cde-trade-ingest.asx.com.au:5XXX/tradeingest   #URL of customer's trade ingest API
```

Connectivity Updates

Connectivity Overview



Connectivity Updates since last Working Group

Ledger API

From CDE 3, based on feedback enumerated types were switched from JAVA classes to native JAVA types.

For example *ReceiveDelivery1Code*

```
<xs:simpleType name="ReceiveDelivery1Code">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DELI"/>
    <xs:enumeration value="RECE"/>
  </xs:restriction>
</xs:simpleType>
```

However for enumerated types with a single and constant value these remained as JAVA classes

- DeliveryReceiptType2Code__1z_FREE
- InstrumentIdentificationType_ASX_1z_INFO
- Issuer_ASX_1z_XASX

Single value enumerated types will be standardised in a subsequent release.

Extractor Overview

Extractor Overview

Extractor

- Provides CHESS Users the ability to stream data from the Node to a client hosted SQL database
- Extractor provides a bridge between the Node and databases hosted by CHESS users or permissioned third parties
- Provides the ability to stream data for a specific party (UIC) from genesis or point in time using an offset
- The Extractor is an application available directly from the DAML SDK (daml.com), it runs on JAVA in a client's internal infrastructure and streams into a Postgres database
- From CDE 4 the CHESS Application will contain a first release of 'API contracts', these API contracts represent the state of data on the Ledger (along with the choices that resulted in any state change)
- State contracts will incrementally become available in each subsequent CDE release

Extractor Overview

Extractor

- All data is streamed into 3 key SQL tables;
  - Transaction
  - Exercise
  - Contracts
- All events are linked by a single transaction ID.
- Data represented in JSON format

Example, the full transfer of units from one HIN to another HIN where no holdings existed would result in;

- A new transaction
- A record of the exercise / choice that resulted in that event
- The archival of the contract where the units were being transferred **from (delivering HIN)**
- The creation of a new contract where the units transferred **to (receiving HIN)**

Key Takeaways

Extractor

- From CDE 4 all developers will be able to access the Extractor
- Data can be streamed into a client's internal Postgres database
- API contracts will be progressively made available from CDE 4
- Multiple connections available to the Node, hence multiple instances of databases
- This model does not preclude developers using the core Ledger API to develop other models / solutions, the Extractor provides a pre-build application for the specific purpose of streaming data

Questions

Next Working Groups

6 November

- Demand Reporting Updates
- Review of Ledger API Contracts examples

4 December

- CDE 4 updates post implementation
- Browser Updates

Thank you

Disclaimer

This document provides general information only and reflects matters put forward for discussion at a point in time. You should obtain independent advice before making any decisions. ASX Limited (ABN 98 008 624 691) and its related bodies corporate ("ASX") makes no representation or warranty with respect to the accuracy, reliability or completeness of the information. To the extent permitted by law, ASX and its employees, officers and contractors shall not be liable for any loss or damage arising in any way (including by way of negligence) from or in connection with any information provided or omitted or from anyone acting or refraining to act in reliance on this information.