



# CHES Replacement Project

**Connectivity & Integration  
Working Group  
Direct (node) Connectivity**

4th December 2018

# Important Information – Competition Law Policy

Working group members are reminded to have regard to their obligations under competition law.

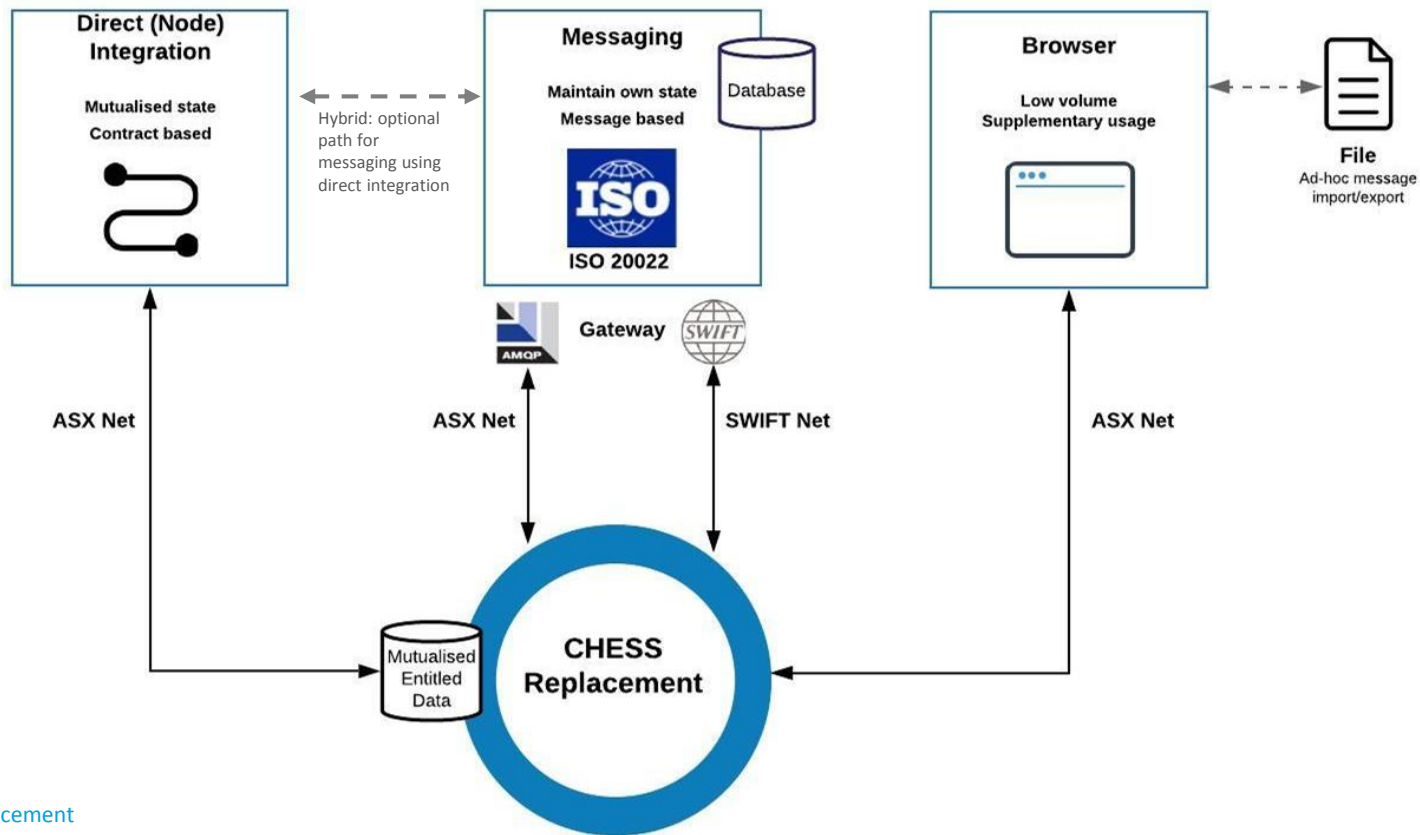
In particular, please note recent changes to the Competition and Consumer Act to prohibit a corporation from engaging with one or more persons in a concerted practice that has the purpose, effect or likely effect of substantially lessening competition.

# Agenda

- > Introduction
- > Digital Asset Overview
- > Transaction Flow
- > Ledger API Connectivity
- > Ledger API Demonstration
- > Data Streaming
- > Direct Connectivity vs Messaging
- > Questions

# Introduction

# Introduction: Connectivity Options



# Digital Asset Overview

# Digital Asset

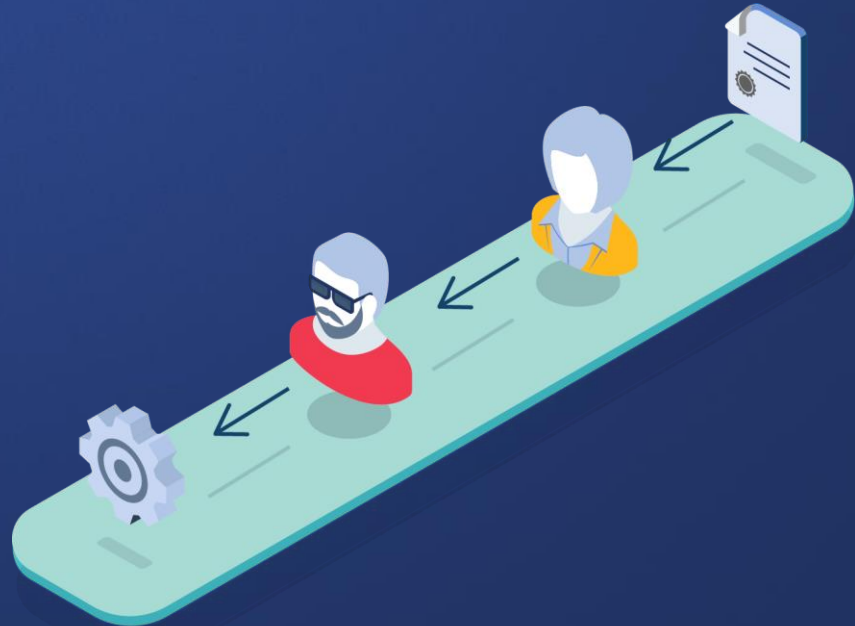
Is a technology company that builds products based on Distributed Ledger Technology (DLT) for highly regulated institutions, such as financial market infrastructure providers, CCPs, CSDs, exchanges, banks, custodians, health insurers and their market participants.

## Quick Facts

- We were founded in 2014 and now have over 180 employees across 6 countries
- We deliver flexible infrastructure for market participants to share processes and data securely, on a need-to-know basis, without the need for reconciliation
- Ours is the only platform being built to the production requirements of major institutions
- We have raised over \$115m from more than 15 strategic investors across the financial ecosystem
- In December 2017, the Australian Securities Exchange announced its intention to replace its post-trade clearing and settlement system with DA technology
- In October 2018, Hong Kong Exchanges and Clearing Limited announced the completion of a successful prototype and are now collecting production requirements with the market.

# What are we solving for?

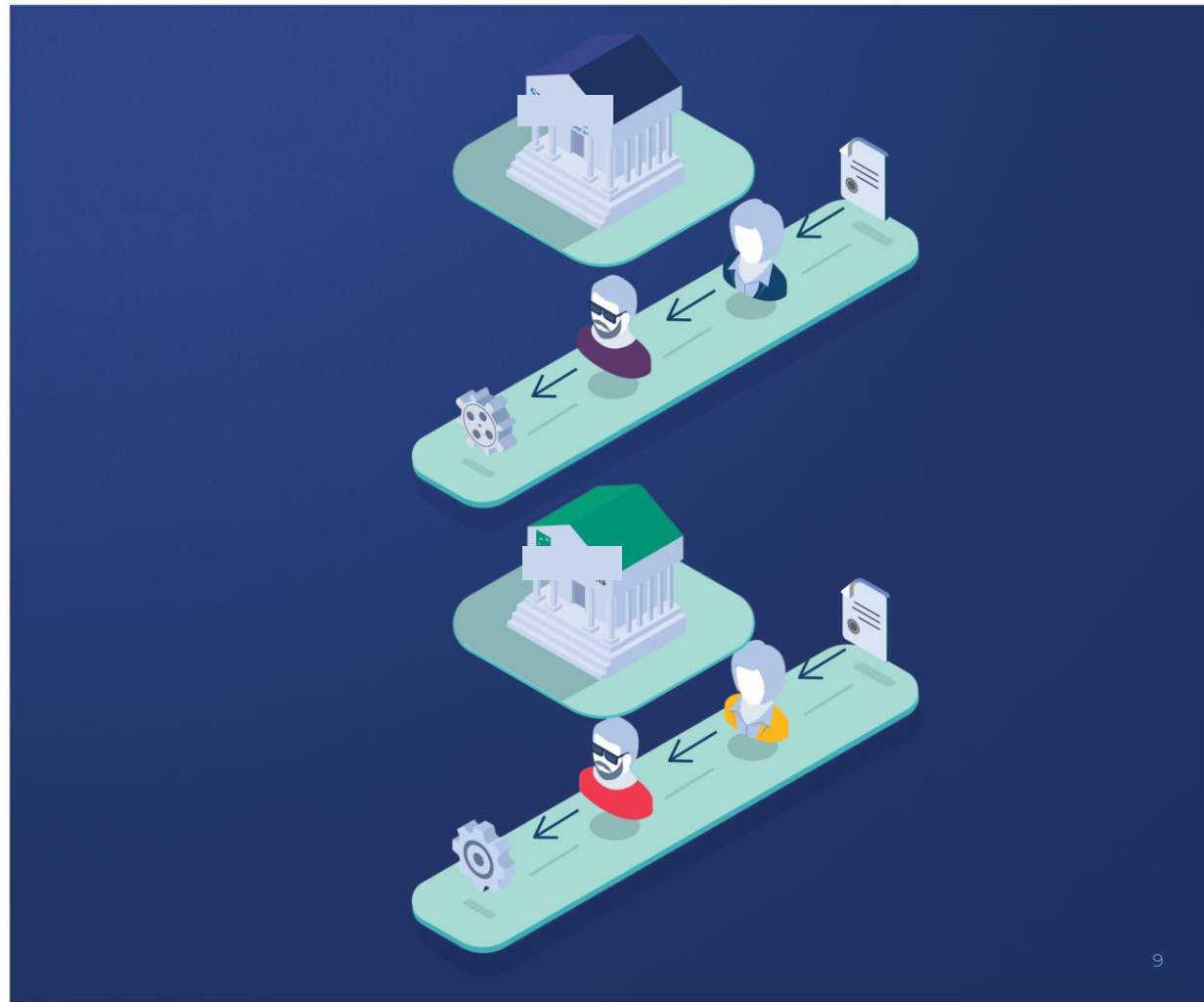
The process of creating software in highly regulated financial markets is a process of **translation** - from legislation / market rules; to legal and compliance teams; to product and engineering; to functioning software systems - **and is expensive, difficult and error prone.**





# Everyone does it differently

Every institution undertakes this translation process independently – **leading to many different solutions to the same problem**, and inconsistencies across the market – there is no common domain model across participants



# Markets rely on multi-party workflows

Participants in a market collaborate in complex multi-party workflows across institutions to deliver services for end investors.

Eg

- Clearing and Settlement
- Pre-settlement workflows
- Corporate Actions
- Securities Lending



# Messaging limitations

Standardised messages passed between institutions do not guarantee consistency of processing by different technical systems across counterparties or provide a common domain model of the problem space. As a result, a long tail of operational cost and risk remains in the system.

Message driven asynchronous systems developed across institutions are not well suited for complex, multi party workflows.



# Technology islands create duplicated cost

Independent, bespoke systems at each institution can be optimised for STP within the institution; but are not easily extended or updated in concert with counterparties or the wider market.

Regulatory driven change forces market wide updates to technology systems – a major, ongoing duplicated cost that does not provide differentiating client value for participants.



# How do we solve for this?

The Digital Asset Platform is designed to provide a common platform for market participants.

1. Continuous data integrity across all market participants
2. Distributed straight through processing across institutions.
3. Extensible and upgradeable functionality for individual institutions or across market through apps.



# Inside a node

## APPLICATION LAYER

Modular and extensible through the development and installation of new apps (like a modern phone)  
Facilitates integrations to existing Systems (eg by SWIFT/FIX/other format)

## DAML BUSINESS LOGIC ENGINE

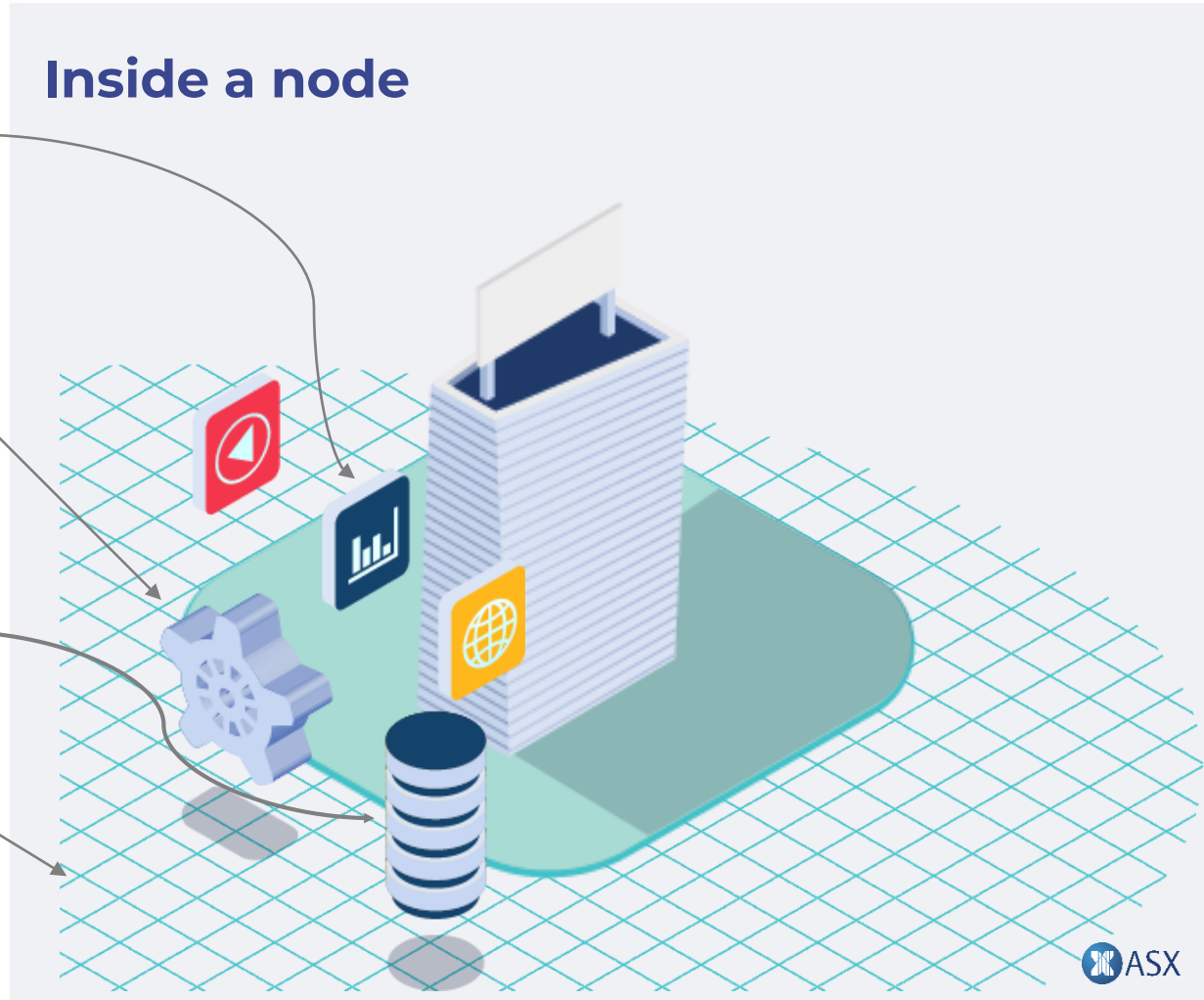
Common across all participants in the market  
Enforces both market rules and contractual rights  
Offers knowable finite set of outcomes  
Built-in safeguards

## PRIVATE CONTRACT STORE

Physically segregates confidential info  
Contracts are Stored locally by parties involved

## GLOBAL SYNCHRONISATION LAYER

Globally synchronized amongst all participants  
Does not reveal info about data itself (proofs only)  
Provides integrity and transparency guarantees



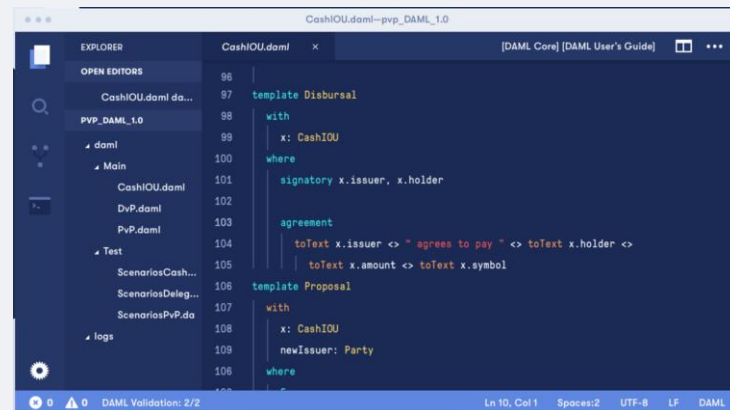
# Transaction Flow

# DAML

- A key element of CHES Replacement is the use of the DA Platform and Digital Asset Modelling Language, called DAML.
- DAML is a means of mapping the rights and obligations across all ASX participants and tracking their changes over time on the ledger.
- DAML guarantees to keep all participants synchronised and protect privacy and confidentiality of information.
- All modifications to the ledger happen via DAML commands, resulting in DAML events. The ledger is updated by a participant upon submission of a valid transaction (DAML command). The rules as encoded in DAML dictate under which conditions contract instances are added/archived.

## DAML Libraries

- ASX DAML contracts will be grouped into DAML Libraries (or modules). Tip: think Java .jar file
- DAML Libraries represent the application rules (i.e. ASX market rules)
- The structure of commands and events are defined by the DAML Libraries being used
- Node users will be provided with the CHES Replacement DAML Library and SDK (Software Development Kit)



```
CashIOU.daml
96
97 template Disbursal
98 with
99   x: CashIOU
100 where
101   signatory x.issuer, x.holder
102
103 agreement
104   toText x.issuer <> " agrees to pay " <> toText x.holder <>
105   toText x.amount <> toText x.symbol
106
107 template Proposal
108 with
109   x: CashIOU
110 newIssuer: Party
111 where
```

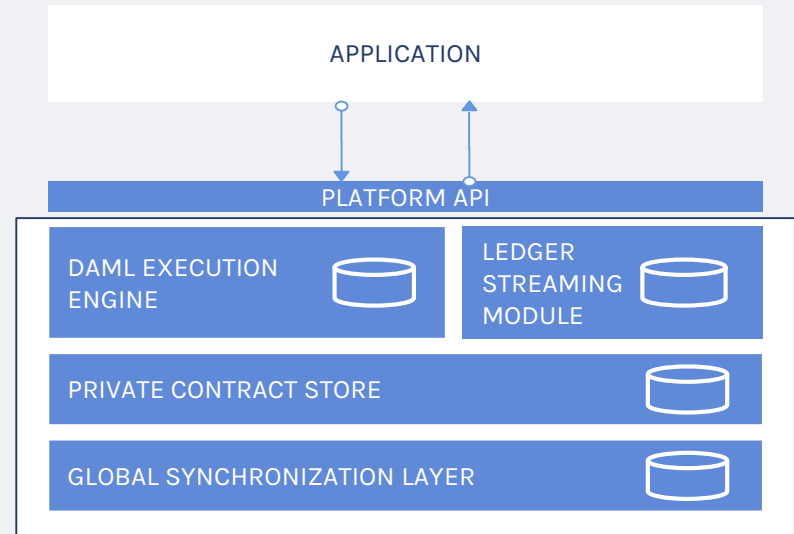


# Direct Node Access

Direct Node Access provides secure and direct connection to the DA Platform. Node access will allow Transaction Submission and Block Event queries of DAML based contracts on the distributed ledger. It maintains the complete state of a financial institution distributed among its participants.

**The DA Platform is a continuously evolving software stack, which gives the following benefits:**

- Eliminates the synchronization process and latencies caused by such a process
- Creates full transparency and provable state of the market
- Enforces market rules
- Handles distribution / persistence
- Constructs DAML contracts
- Performs validation
- Creates blocks on the blockchain and manages notifications



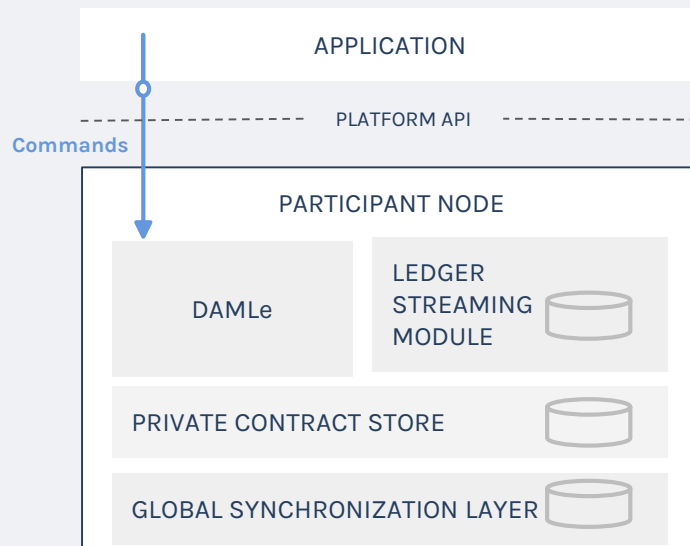
# Direct Node Access

All modifications to the ledger happen via DAML commands. The DA ledger is updated by a participant when it submits a valid transaction command. The rules as encoded in DAML dictate under which conditions contract instances are added/removed.

## A valid transaction command consists of:

- Type-checked and interpretable DAML update expression
- The Business Intent Message Commitment, which is the blinded hash over the customer-specific intent behind a command
- Maximum Record Time
- Ledger Effective Time

With the DA Time Model, the Application always has the ability to determine when a command is rejected, accepted, contains errors, or if a command gets lost. The structure of commands and events are defined by the DAML Libraries being used.

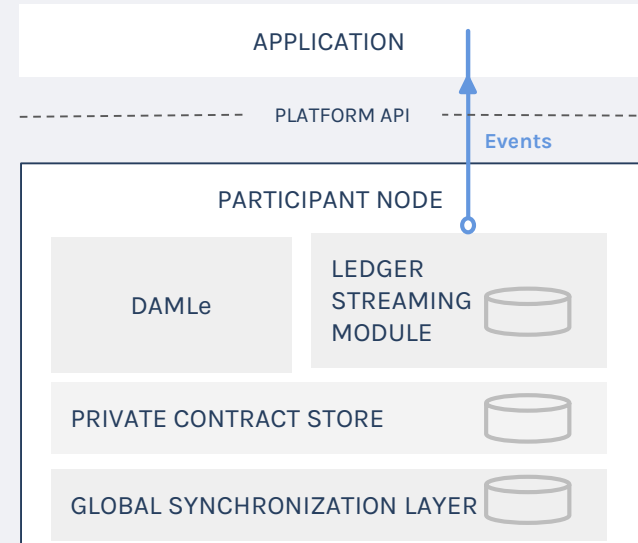


# Reading via block-event queries

Transaction events are grouped in blocks. A transaction event contains an optional reference to the business intent message (BIM) that caused it, the ledger effective time of this transaction, the index within the block, and a list of events.

## Block Events:

- A transaction event is the atomic building block of the transaction. It either creates a new contract or archives an existing one
  - Create event: contract ID + template ID + parameters
  - Archived event: contract ID + template ID
- Microservices in the Application Frameworks are listening to events to trigger automation processes
- A block also contains a possibly empty list of errors that occurred during the building of this block



# Ledger API Connectivity

# Why gRPC?

- Open-source, with many adoption examples and a vivid community of contributors
- Uses **protobuf**: a language & platform agnostic serialization format
  - Does not tie to a specific technology stack
  - Binary representation for wire transfer, designed for scalability and high performance
  - Optimized for low network footprint
  - Understood by off-the-shelf infrastructure components such as firewalls, load balancers, proxies etc.
  - Multitude of language mappings - simple generation of idiomatic language bindings
    - proto files are composable schema definitions
    - protoc (plugins) to generate bindings to all major ecosystems
    - SDK provides JAVA and Python client libraries.
- Built on top of HTTP2 - goes beyond the rigid set of verbs offered by HTTP
  - Supports both uni- as well as bidirectional streaming
- Authentication in the form of SSL/TLS and custom pluggable mechanisms
- Designed with simplicity in mind
  - Simplifies SDK deployment - just compile source files and copy sample code

# Exploring the Endpoints

## Administrative Services

- **LedgerIdentityService**
  - Returns the ID of a given ledger
- **LedgerConfigurationService**
  - Streams server-side configuration changes
- **PackageService**
  - Returns information about packages loaded onto ledger

# Exploring the Endpoints

## Command-related Services

- **CommandSubmissionService**
  - Submits commands to the ledger
- **CommandCompletionService**
  - Streams completions (OK / Error) for command submissions
- **CommandService**
  - Higher-level abstraction
  - Allows for the submission and subsequent completion of a command in a single call
  - Implements throttling (max requests in flight)

# Exploring the Endpoints

## Event-related Services

- **TransactionService**
  - Streams transactions containing ledger events
  - Streams transaction trees (contracts + provenance)
  - Allows transaction lookup
  - Allows filtering by template
  - Sandbox allows multi-party stream
- **ActiveContractService**
  - Streams only events required to build up ACS
  - Allows quick re-hydration
  - Suggested pattern is to catch up using ActiveContractService, and then subscribe to TransactionService with an offset to keep ACS up to date



# Towards Application Frameworks

## Capabilities

- Reactive / Stream library support
- Command Retry
- Command Backpressure / Throttling
- Event stream recovery
- In-memory contract store
- Persistent contract store
- High-Availability / Recoverable contract store
- Pending contract handling
- Strongly-typed contracts (code gen)

# General Workflow for Direct Connectivity

## What is a Command?

- A command is the exercise of a choice on the participant's Master Contract, eg, a command to open an account at CHES Replacement.

## What is an Event?

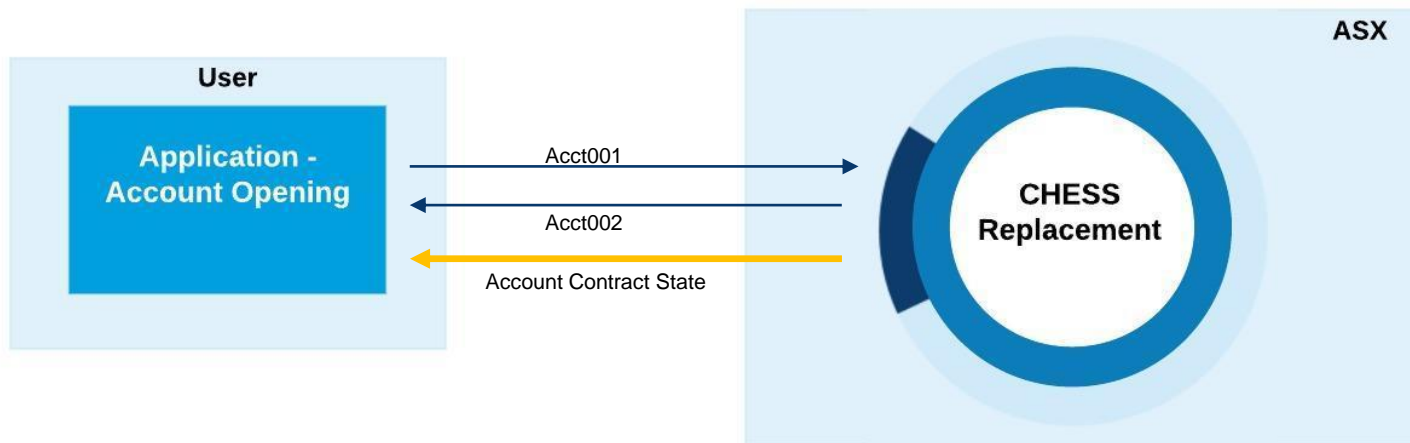
- An event is the resulting contract(s) creation(s)/archival(s) as a consequence of a command submission.

## What is the general workflow for an ASX Participant to connect directly with CHES Replacement ?

- Open a gRPC channel and connect to the ASX CHES Replacement Ledger
- Download the DAML package containing the template for the participant's Master Contract
- Subscribe to the Transaction Service to receive transactions (which contain Events - the results of Commands)
  - Receive the Master Contract creation event on the Transaction Service stream.
- Send commands to exercise choices on the Master Contract, using the Command Submission Service.
  - Command Completions can be observed on the Command Completion Service
  - Resulting events (eg, contract creations/archivals) can be received on the Transaction Service stream

# Ledger API Demonstration

# Transaction – Account Registration Request



Message Gateway: Ledger API

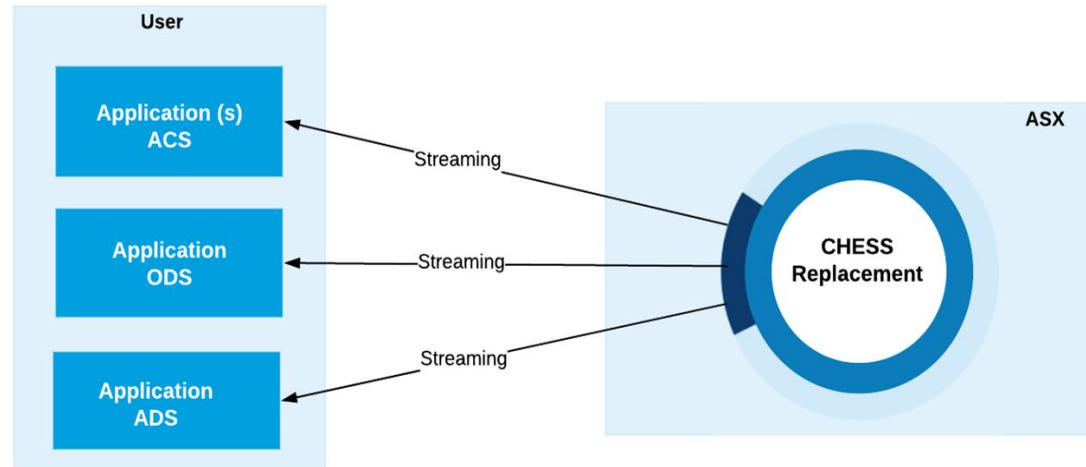
Transaction Format: DAML

State: Managed on ledger

# Data Streaming

# Data Streaming

Direct connectivity provides data streaming capabilities directly from the Ledger Event Stream (part of ledger API)



## Active Contract Set (ACS)

- Current State
- Vendor / Participant developed application(s)

## Operational Data Store (ODS)

- Current State and History
- ASX developed application

## Analytical Data Store (ADS)

- History
- ASX developed application

# Direct Connectivity vs Messaging

# Direct Connectivity vs Messaging

	Ledger API	AMQP	SWIFT
Supports CHES Replacement ISO 20022 instruction set	✓	✓	✓
Multifactor security	✓	✓	✓
ASX Net (VPN)	✓	✓	✗
SWIFT Net	✗	✗	✓
Access real time golden source data	✓	✗	✗
Extensible to future products and new applications	✓	✗	✗
Mutualised multiparty workflows	✓	✗	✗
Flexible infrastructure for regulatory and market change	✓	✗	✗



Questions?

Thank you

# Disclaimer

*This document provides general information only and reflects matters put forward for discussion at a point in time. You should obtain independence 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.*

*© Copyright 2019 ASX Operations Pty Limited ABN 42 004 523 782. All rights reserved*