

Delivery Framework Documentation

version Master

R3

July 09, 2018

Contents

Welcome to Corda Solutions!	1
Target Audience	1
R3 Delivery Framework	1
Delivery Framework Introduction	1
Overview	1
History	1
Principles	1
Sections	2
Commercial Framework	2
Introduction	2
Overview	3
Principles and good practices	4
Practice / Step 1	4
Practice / Step 2	5
Practice / Step 3	5
Templates	5
Examples	5
Partner Selection	5
Overview	5
Corda Partner Network	6
Principles and good practices	6
Initial Partner Scoping	6
Partner Shortlist Creation	7
Partner Engagement	8
Partner Selection and Onboarding	8
Business Case	9
Overview	9
The Business Case for Blockchain	9
The Business Case for Corda	9
Principles and good practices	10
Terminology and Formulas	10
Introduction	10
[Per Target]	10
Outputs	10
Activities	10
Inputs	10
Gross Profit	11
Capital Activities	11
Net	11
Executive Summary	11

Examples	11
Product Design and Build	11
Overview	11
Principles and good practices	11
Principle 1 - Determine the Project Sponsors	11
Principle 2 – Role of Product Owner	12
Principle 3 – Customer Driven Design	12
Principle 4 Architecture Design	12
Principle 5 - Validate with Customers	13
Principle 6 - Platform Requirements	13
Principle 7 - Integration Architecture	13
[R3 INTERNAL] Legal	13
Overview	13
Principles and good practices	14
Principle / Practice 1	14
Principle / Practice 2	14
Principle / Practice 3	14
Principle / Practice 4	15
Principle / Practice 5	16
Principle / Practice 6	16
Principle / Practice 7	16
Principle / Practice 8	16
Principle / Practice 9	17
Principle / Practice 10	17
Examples	17
[R3 INTERNAL] Legal and IP Framework	17
Overview	18
List of Legal/IP Artifacts	18
History	18
Principles and Good Practices	18
Principle 1 - Determine Type of Engagement	18
Principle 2 - Determine Services Supplied	19
Principle 3 Determine Key Parties	19
Principle 4 - Determine Corda Version	19
Principle 5 - Determine IP Requirements	19
Pre Existing or Contributed IP	20
Developed IP	20
Non Exclusivity	20
General Terminology	20
Terms Specific to R3 Corda Software License Agreement	20
License Grant	20
Ownership Rights	20

Restrictions on Use	21
Confidentiality and Use	21
Terms and Termination	21
Templates	21
Examples	21
[R3 INTERNAL] Regulatory	21
Overview	21
Principles and good practices	22
Principle / Practice 1	22
Principle / Practice 2	22
Principle / Practice 3	23
Principle / Practice 4	23
Principle / Practice 5	24
Principle / Practice 6	24
Principle / Practice 7	25
Principle / Practice 8	25
Examples	25
Go To Market	25
Overview	25
Principles and good practices	25
Principle / Practice 1	25
Principle / Practice 2	26
Principle / Practice N	26
Templates	26
Examples	27
Corda Business Networks	27
Introduction	27
What is a Business Network?	27
What are the Challenges Associated with Business Networks?	27
How Do Business Networks Start?	28
Top-Down (Business-Led)	28
Bottom-Up (Technology-Led)	28
Roles and Responsibilities	28
Business Network Governor (BNG)	28
Business Network Operator (BNO)	28
Business Network User (BNU)	29
Successful Business Network Deployment (The Playbook)	29
S. Structure	29
S.1. Define Business Network Mission	29
S.2. Assign Business Network Roles	29
S.3. Define the Governance Body Model	29
G. Governance	29

G.1. Describe Participants	30
G.2. Define the Asset Model	30
G.3. Outline Legal and Regulatory Needs	30
G.4. Establish the Network Financial Model	30
G.5. Establish Data Privacy Policies	30
G.6. Establish Data Retention policies	30
G.7. Establish Data Resiliency and System Availability Policies	30
G.8. Assert Data and/or Processing Standards	31
G.9. Establish Transaction Validation Policies	31
G.10. Establish Dispute Management Policies	31
G.11. Establish On-boarding Policies	31
G.12. Establish Off-boarding Policies	31
G.13. Establish Inter-Network Policies	31
G.14. Define Performance SLAs	31
G.15. Establish Change Management Policies	31
G.15.1. Business Network Role Changes	32
G.15.2. Business Network Legal Agreement Changes	32
G.15.3. Corda Platform Change Management Rules	32
G.15.4. CorDapp Change Management Rules	32
O. Operations	32
O.1. Devise Node Configuration	32
O.2. Notary Configuration	32
O.3. Oracle Configuration	32
O.4. Create change management procedures	32
O.5. Establish Operational Times	33
O.6. Establish monitoring metrics	33
O.7. Define audit, report and documentation requirements	33
O.8. Define the network support parameters	33
O.9. Define Data Retention and Pruning Procedures	33
Architecture	33
Engineering	33
Foreword	34
Data model	35
Data privacy	35
Data updates	37
States	37
Contracts	37
On-chain vs off-chain	37
Contracts	38
Signers vs Participants	38
Determinism	39
Flows	39

Checkpointing	40
Corda Services	40
Flow upgrades	41
CorDapp structure	41
Contracts and States	41
InitiaitingFlows and InitiatedFlows	41
Build	42
Operations	42
Frequently asked questions	42
What is this site for?	42

Welcome to Corda Solutions!

What does it take to build a solution to a distributed problem using Corda? This seems like an easy question, but it's trickier than might first appear. Distributed and, in particular, decentralised problems typically need to address questions around how all of the participants come to agreements, and this needs more than just code to achieve it.

Corda provides the technical capabilities we need to build these sorts of solutions, but how should we think about these other issues? When we have thought about them, then are there specific technical approaches that we should use?

This site aims to ask some of these questions and offer some solutions.

Target Audience

talk about

- an open source distributed application delivery framework -
- what is this thing
- why is it valuable
- who is it for (different roles / different focus)
- platform info (corda.net)
- where to get more information (SALES / PROFESSIONAL SERVICES / PARTNER)
- phases of your development lifecycle / journey
- sample apps - links to marketplace

R3 Delivery Framework

This is the delivery framework root.

Delivery Framework Introduction

Overview

Distributed Ledger Technology (DLT) has brought about completely new decentralized top of stack applications and business models. R3, as one of the first and largest DLT consortiums in the world, has garnered an immense tool kit of best practices in ideation, development, and commercialization of these applications.

History

A framework is a methodology that provides a systematized way of approaching a common business problem. By using frameworks, one can conduct client engagements in a quicker and more standardized manner. Some frameworks may address specific use cases that are already identified, while others may start with a high-level view and then dive deeper.

The R3 Services Delivery Framework is intended to be a tool kit of principles, best practices, lessons learned, and templates compiled by Delivery Managers who have worked on over 90+ projects since the inception of R3 ecosystem.

Principles

- Customer first
- Reuse existing collateral and best practices when possible
- Establish sponsorship and accountability from the beginning
- Always be a Corda evangelist

- Leverage our partner and membership ecosystem

Sections

The following sections comprise the R3 Delivery Framework

- Partner Selection
- Commercial Framework
- Business Case
- Product Design and Build
- Legal and Regulatory
- Legal and IP
- Go To Market

Commercial Framework

Introduction

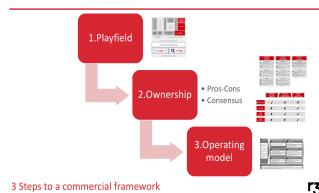
In 2016, the design concepts for Corda were described for the first time in Mike Hearn's technical whitepaper https://docs.corda.net/_static/corda-technical-whitepaper.pdf. It was to be decentralised global database, providing a platform for decentralised application development.

The paper describes the concept of "CorDapps" (Corda Distributed Applications) and how these might be constructed, but did not attempt to describe very much more detailed thinking.

Building and managing distributed applications is, however, more involved than just creating software: In a CorDapp, information and/or assets are exchanged, or transacted, between participants because they represent something meaningful to each of them, but how are those meanings determined, on what basis, and by whom? We address this in the CorDapp Business Network section.

In a nutshell: Business networks for specific objectives comprising three roles/activities;

- i. Governance
- ii. Operations
- iii. Usage



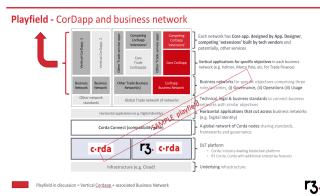
Further, increasingly CorDapps are starting to have 'CorDapp extensions'. Even though the word CorDapp can denote a top of the stack application, we are seeing these CorDapps emerge as platforms in their space and allowing others to build applications on top.

In other words some CorDapps have increasingly explored the opportunity to become a middleware across their playing field. In such a case 'CorDapp Extensions' become the top-of stack user-facing application. e.g. <TBD:insert generic description of MP and example of another potential-app build on top of TIX platform/APIs>

To summarise overall there could be 4 roles;

- CorDapp Business Network Governor
- CorDapp Business Network Operator
- Core CorDapp
- CorDapp Extensions

It might be that one-entity might play more than role among these.



Overview

The importance of the Commercial framework topic is evident in the fact that corporate governance and the ownership structure of companies and initiatives is currently characterised by change processes as the economies of the world becomes more and more globally integrated.

Commercial framework and ownership structures are also of major importance in corporate governance because they affect the incentives of ecosystems, participants, managers, and thereby the efficiency of firms.

The structure is also characterised by the distribution of equity with regard to votes and capital, but also by the types of the equity owners. A classic reference is Jensen and Meckling (1976), who discussed the nature of agency costs associated with outside claims on the firm.

The increased volatility of corporate commercial portfolios observed in recent years has led to renewed interest in ownership framework structures, especially with respect to multinational enterprises.

As the economies of the world become more and more distributed as well as globally integrated, such issues will become more prominent and will affect our understanding of the interweaving systems of corporate relations, through which formal and invisible networks of power are established (Heubischl, 2006).

Commercial frameworks are an important means for governance as well as power structures. Likewise, for interlocking directorships, share ownership may provide influence and control over a third party. Recently there has been considerable interest in the corporate governance practices of modern corporations and it is better to be cognizant of the choices available as well as the impacts as early as possible.

Superficially, our problem is not particularly new. Companies and consortiums have brought people together to transact value since centuries! Our target problem here is a group, or network, of independent parties that want to use Corda to transact something of value to each, and as we're transacting something valuable then this almost inevitably implies a commercial, or business, purpose.

3 broad constructs of ownership exist:

1. A Commercial Software Company (e.g. Microsoft, SAP, Oracle) operates to:
 - Function: Create a commercially driven leading software company with a superior offering (and other services)
 - Objective: Maximise economic value for the software company and its shareholders
2. An Exclusive Consortium of banks/others (e.g. Star Alliance, One world):
 - Function: Create a commercial/ non-profit exclusive consortium of banks/ others who own and manage the application(s) & business network
 - Objective: Maximise value (economic or other) for each member of the consortium
3. A Universally shared industry utility (e.g. TCP/IP, SWIFT):
 - Function: Create a utility that is shared industry-wide for all ecosystem participants
 - Objective: Maximise adoption of the utility by the entire ecosystem, with first-mover advantage

As one can see from above the objective of maximisations are very different among the 3 options outlined: it can be observed that a 'Commercial software company' and 'Universally Shared model' lie quite apart on the spectrum. To aid the decision in above one can see 5 factors:

- Commercial Model
- Economic Beneficiaries
- Access to Solution
- Ownership
- Economic Model

1. Commercial software company: Commercial excellence i.e. Attractive value proposition (solution, pricing etc.) delivered with a focused sales, delivery & servicing model
2. Exclusive consortium of corporate/FIs: 'Winning consortium' i.e. Speed and effectiveness of attracting the top Trade banks and creating network effects
3. Universally Shared industry Utility: Industry alignment i.e. Broader industry alignment and adoption of VoltronX standards, facilitated by regulatory endorsement

Typically, below are the 3 broad constructs of ownership with 8 choices, to further the processes while choosing the appropriate model:

1. Commercial software company:

- Joint Venture: A new for-profit entity with shareholder funding and board membership. Here typically the JV or the new firm owns the IP, selects members of business network and manages the platform
- Technology Partner: CorDapp is owned and maintained by the technology partner, and is licensed out to FIs/Corporates and other stakeholders in the CorDapp business network
- Single corporate/FI: CorDapp is wholly owned and managed by a single corporate/FI. Owner holds IP and exclusive right to provide access to other banks and service providers on the platform

2. Exclusive Consortium of corporates/FIs:

- Joint Venture: New for-profit entity to maximize value to members with shareholder funding and board membership. JV / new firm owns the IP, selects members of business network and manages the platform
- Foundation: A non-profit entity with a select set of members providing funding and given board membership. The Foundation owns IP, selects new members and manages the platform
- Closed Co-op: Corporate/FI consortium founded Co-op with specific membership requirements, Voting rights & profit sharing (if any) based on contribution of each co-op member (e.g. Star Alliance)

3. Universally shared industry utility:

- Foundation: A non-profit entity with all members providing funding and given board membership. Entity owns IP, selects new members and manages the platform
- Open co-op: Co-op of banks & non-banks which jointly manages IP and aims to get new members to join. Voting right and profit sharing based on contribution of each co-op member (e.g. SWIFT)

With above overview, below principles and practices would help in the above choices as well as setting up the commercial framework for your CorDapp!

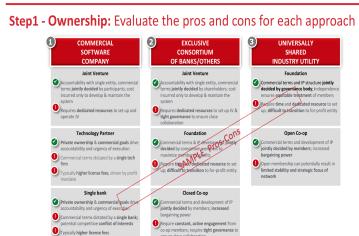
Principles and good practices

All the 4 roles of 'CorDapp Business Network Governor', 'CorDapp Business Network operator', 'Core CorDapp' and the 'CorDapp extensions' would have potentially different choices among commercial-models. A coherent consensus among the choices by all the 4 role-players will be necessary for everyone to come together and transact. #this is the main 'content' section, attempt to break down each view into a series of principles and/or practices and describe them...

Practice / Step 1

Evaluate the pros and cons for each (among 8) approaches for Ownership

This needs to be done by each of the 4 role-players individually. A really simple template of this can be as below attached.



Practice / Step 2

Reach consensus among the stakeholders and establish the ownership approach

Each stakeholder share their preferences and reach consensus among them. A really simple template of this can be as below attached.

Step2 – Ownership: Reach Consensus among different roles

	1 COMMERCIAL SOFTWARE COMPANY	2 EXCLUSIVE CONSORTIUM OF BENEFICIARIES	3 UNIVERSALLY SHARED INDUSTRY FACILITY
Software Provider	✓	X	X
Core Supplier	X	X	✓
Certified Network Operator	X	✓	✓
Other Business Network Operator	X	X	✓

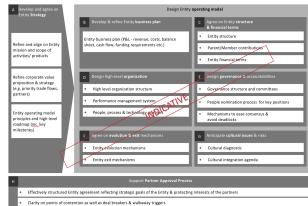
r3

Practice / Step 3

Finalise the Operating model by setting up the entity.

A really simple template of this can be as below attached.

Step3 - Operating model: Setup the entity



r3

Templates

Include any document templates here. Link to the source file in the master location. NO POWERPOINT, ever.

Document	Purpose	Owner	Last Updated
Filename / link here	Short text description	Owner name here	Date here
Filename / link here	Short text description	Owner name here	Date here
Filename / link here	Short text description	Owner name here	Date here

Examples

Include any client examples here to make it more real. Any examples should be uploaded and linked to in PDF format – always mark files with the IP licence attached (i.e. r3 internal, r3 consortium, public).

Document	Why its good	Key lessons	IP / Distribution Rights
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal

Partner Selection

Overview

Welcome to Corda Solutions!

The Partner Selection framework is an important element of R3's delivery framework as it enables a consistent and scalable approach to connecting consumers and producers and supports the build out of an ecosystem for Corda solutions.

Types of Partners & scope for this framework include:

- Technology partners providing key infrastructure, platforms and services for Corda and clients
- Systems integrators (SI) delivering proof-of-concepts, pilots and production solutions for clients on Corda
- Consulting firms advising their clients on the business potential of DLT and Corda
- Software vendors (ISV) building end-user applications on Corda
- Network Service Providers

Corda Partner Network

Over the past 18 months R3 have build up an extensive partner list as shown below.



Principles and good practices

The Partner selection framework can be split into four distinct components:

- Initial Partner Scoping
- Partner Shortlist Creation
- Partner Engagement & Partner Selection and Onboarding

Initial Partner Scoping

Typically, the R3 Services Delivery Manager for a project should work with project participants (the clients) as the single point of contact to define the project's partnership needs & selection criteria i.e. given the CorDapp value proposition, what partner type/s are required to fulfil the roles in the Business Network.

Besides the core actors transacting in the network, typical roles include:

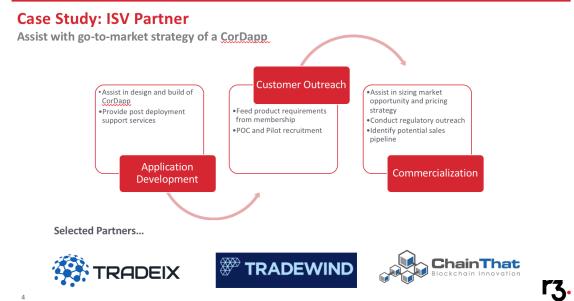
- Business CorDapp Builder
- Business Network Designer
- Business Network Operator
- Business Network Governor

There may be more roles required and these can be fulfilled by a single or multiple entities. Note: See the Business Network section of the Delivery Framework for more detailed descriptions of these roles.

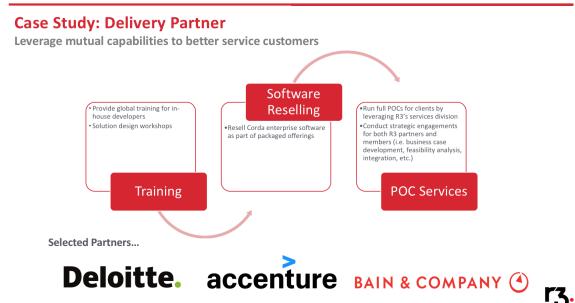
The maturity stage of a project should be considered before initiating the Partner selection process i.e. is it right time to engage, is there clear client demand for the proposition and intent to support any resource/funding required for this as a commercial product. Note that if the demand for a partner is initiated from a outside of the R3 Services project portfolio, that group should engage directly with the R3 Partner team via partner@r3.com.

Below are some case studies which help demonstrate the various ways in which R3 can engage with our partners depending on the use case.

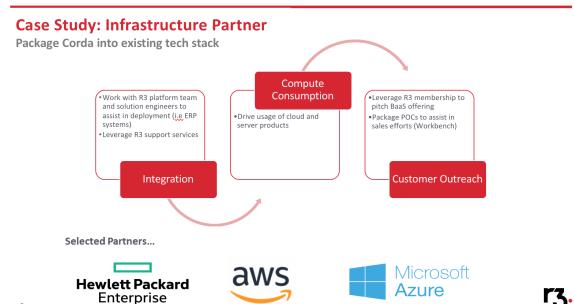
ISV Partner



Delivery Partner



Infrastructure Partner



Partner Shortlist Creation

Creating a Partner shortlist

Typically, the R3 Services Delivery Manager for a project should engage the R3 Partner team to provide a project briefing and request for a partner (via partner@r3.com), who will then assign a single point of contact for the opportunity. The R3 Partner contact will determine an initial appropriate partner shortlist based on 4 key criterion:

- Corda Proficiency
- Domain expertise
- Geographic footprint
- Reputation and track record of successful delivery

Once this initial partner shortlist has been refined & agreed with project participants, the R3 Partner contact will provide introductions to key contacts at shortlisted partner organisations to the R3 Services contact for the project. The R3 Partner team will typically source partners from R3's existing partner network. In the instance where the R3 Services team (on behalf of the clients) want to shortlist a partner not in R3's existing network, the Partner team will first lead the onboarding of that Partner into the R3 partner network.

Leveraging the R3 Ecosystem

The R3 Ecosystem (<http://ecosystem.r3.com/>) is a flourishing community that brings R3 partners and members together with a shared focus on moving blockchain applications into commercial deployment, and changing finance and commerce from within.

There are a number of relevant R3 Ecosystem components that can be used in relation to Partner selection:

- Discoverability – Contains a Partner Directory, finding out who is in R3's Partner Ecosystem.

- Services Marketplace – Contains information on specific Partner offerings based on client needs for a solution.
- Solution Explorer – Contains existing Partner demos & leveragable components.

Note that the R3 Ecosystem is a public website and will not be the full content or nature of the relationship with a partner. It is recommended to be used as an initial tool in shaping a partner shortlist but is not intended to replace the role of the R3 Partner team in the process.

Partner Engagement

Secure NDA with shortlisted Partners

Partners in R3's existing network will have NDAs in place already (as part of their on boarding). For any shortlisted partners not in the R3 Partner network, an NDA should be secured before any initial engagement is held. Typically, the R3 Legal team (legal@r3.com)

Engage & brief shortlisted partners

The R3 Services Delivery Manager should lead project briefing sessions with shortlisted partners, potentially with the support of a lead client (if established).

The project briefing should provide a comprehensive overview of the project goals, scope and proposed partner role. Clear expectations should also be set at this stage regarding the scope and timelines for shortlisted partners to pitch for the work and provide documented proposals back to the project group (unless another agreed output is agreed). Providing as much clarity here will typically ensure more consistent partner proposals that can be evaluated. If there is significant disparity between the partner proposals received at this stage, some follow clarifications with the potential partners should take place ahead of the selection process.

Partner Selection and Onboarding

Determine decision approach

The R3 Delivery Manager should work with the project participants to agree a lightweight decision approach in order to evaluate partner proposals and make an ultimate decision on a partner. Typically this involves an agreed scoring criteria for partner proposals, with each project participants (& potentially R3) receiving a vote in the process. Some example projects in the R3 Services portfolio that have successfully selected a partner include:

- Project Marco Polo – CorDapp builder partner
- Project Voltron – Business Case partner
- Project Euro Debt - Legal & Reg partner

Note that traditional RFP routes for partner selection are considered out of scope for this process, although this selection route can be explored at this stage if project participants prefer.

Run chosen Partner selection process

The R3 Services Delivery Manager should facilitate the chosen partner selection process & establish clear decision points and inputs required at each stage.

Communicate decision to all parties

The R3 Services Delivery Manager should facilitate the partner selection outcome to both the chosen partner and any unsuccessful partners.

Partner on boarding

If this hasn't been discussed and if required, commercial agreement discussions with the chosen partner should commence at this stage.

Once an agreed commercial agreement has been established, all required legal agreements to govern the overall relationship should be put in place at this stage.

Highlighted examples of successful R3 Partnerships

Finastra

Finastra and R3 Partnership
Fusion LenderComm digitizes communication with lenders – driving efficiencies in the process, saving agents time and money, and eliminating operational risk.

Highly secure nodes on the Fusion LenderComm network maintain all transaction history. This gives every lender a personal view of deals they participate in and a time-stamped audit trail.

Developed in close collaboration with some of the world's top global banks including BNP Paribas, BNY Mellon, NSBC, ING and State Street, Fusion LenderComm is a platform for the syndicated lending community which is underpinned by Corda, R3's powerful distributed ledger technology.

Fusion LenderComm, powered by Corda:

- Seamless collaboration between agent and lenders
- Fully automated, secure communication with lenders
- Real-time data
- Cloud-based technology for quick and easy adoption

FINASTRA

HSBC BNP PARIBAS BNY MELLON ING

STATE STREET

r3.

Calypso

Calypso and R3 Partnership
BBVA, BBVA, and Bancomer deploy DLT-based FX matching pilot with Calypso and R3.

CALYPSO

BBVA, BBVA Bancomer, and Bancomer deploy DLT-based FX matching pilot from Calypso Technology and R3

OCTOBER 2017

"Calypso sees 'distributed ledger' technology as an important component of our innovative offerings to transform the way trade processing operations. We are excited that after a successful completion of the initial pilot, BBVA, BBVA Bancomer and our strategic blockchain partner R3, have engaged in next phase of Calypso FX Matching Distribution Ledger Technology service supported by our Cloud services".

- M. Mayank Shah, Head of Strategy, Marketing and Alliances

BBVA and BBVA Bancomer are piloting an FX matching application developed by Calypso Technology, a leading provider of cloud based cross asset trading software solutions for the financial markets, and enterprise software firm R3. The application, which runs on R3's Corda platform and is facilitated by Calypso's cloud services, has already been used by the two banks to match a test FX trade.

r3.

Business Case

Corda is designed for the future of business. The applications and business networks supported by Corda will succeed because, and only if, the solutions deliver net-positive business outcomes for the users, providers, and necessary sponsors. Therefore, the business case for each business network and CorDapp is an important pillar of a successful solution on Corda.

Overview

A business case is a document that describes and quantifies the relationship between entities and operative parts of the business. Every business is a system with inputs, outputs, and participants, all of which change over time. The business case describes how that system works and shows the viewer with quantitative data.

The Business Case for Blockchain

Blockchain technology may create opportunities to both reduce cost and increase revenue. Marginal improvement is only visible by first quantifying the costs and revenues in the market as-is. This usually consists of a three-level business case: (1) the revenue and cost for end users, (2) the revenue and cost for service providers, and (3) the revenue and cost for platform or supporting technology providers. This as-is picture should be calibrated and customized based on the user or provider's operational capabilities, partners, and context.

The business case for blockchain should target points of centralization, redundant record-keeping, inter-party trust, or time buffers in a process. These can be opportunities to cut cost by removing a middleman, reducing multiple data silos, or reducing cycle and process times. These could also be opportunities to increase revenue by creating new, efficient marketplaces, supplying trust as a service, or supplying revenue-generating activities with more ammunition through efficient secondary (or tertiary) markets.

The Business Case for Corda

With a positive business case for a blockchain technology solution, Corda can support and/or improve the business case given a few characteristics:

1. Privacy among users - If users require anonymity inside the business network, such that other participants cannot infer the identities or activities of other users
2. Scalability of relationships - If users create and use relationships to many other users in varying permutations (A-B, B-C, A-C, A-B-C)

3. Interoperability between applications - If users want to take assets or obligations from one business network and transact that asset on a second (or third) business network.
4. Regulation exposure - If users require the ability to expose parts of transactions to a regulator without exposing unnecessary data or drastically increasing operational risk

Principles and good practices

A business case for a CorDapp and business network can be completed by the participants in a project, by the developers or funders of an application, by the business network operator, or by an outside consulting firm.

In many CorDapps and business networks, there will be two levels of business case: one for the end user of the application and another for the business network operator (or application developer, whoever is on the supply side). For a CorDapp and business network to be successful, the business case must work for both sides. The business case for R3 is not covered in this page.

This page describes the process and desired outcome for a business case, regardless of who delivers it.

The goal of each part of the business case is to break down the part of the system under analysis to a simple “P*Q” level. In the quantity for all parts, use privately-supplied forecasts, public information, or assumptions, in that order of preference. The X-axis of the business case is usually time. If this is the case, pick a timeline that makes sense for the business and the information available. The Y-axis is usually for the list of inputs, activities, and outputs.

Terminology and Formulas

Price = Price of the good/service to the client

Cost = Cost of the input/activity to the target

COGS = Cost of Goods Sold = (Cost of Inputs) + (Cost of Activities)

PTI = Pre-Tax Income

EBIT = Earnings Before Interest and Taxes

EBITDA = Earnings Before Interest, Taxes, Depreciation, and Amortization.

Introduction

Describe the business with participants, inputs, activities, and outputs. Map the business case against the participants and activities. Which participants' business activities are you describing, and for what purpose? Create a new business case for each target. List all baseline assumptions.

The Sections below are divided into operational activities (inputs, activities, and outputs) and capital activities (capital expenditure, long-term investment, appreciation/depreciation/amortization, etc). Operational activities have a marginal impact and can be started or stopped immediately – usually on a 30-day basis. Capital activities usually have an impact over at least three months, or a significant share of the operational activities.

[Per Target]

Outputs

This is the primary revenue part of the business case. List the target's outputs. These are usually products or services. List the options for revenue generation. These include subscription fees, license fees, per-transaction fees, one-off fees, revenue share, etc. As always, the goal is to simplify the business to a “P*Q” level. Also note whether you are describing actual sales, revenue recognition, and any relevant accounting treatment.

Activities

This is the first cost part of the business case. List the targets' activities and how the target pays for these, again on a “P*Q” basis. Cost options for activities could be per [time], per input, per output, etc.

Inputs

This is the main cost part of the business case. List the target's inputs and how the target pays for these. Include all inputs required for the activities and outputs. Describe the cost for these inputs based on how the target pays for these

Gross Profit

Gross profit is assessed on an operational basis. $GP = (\text{Revenue}) - (\text{Cost})$ over the relevant period of time. $GP\% = (GP)/(\text{Revenue})$

Capital Activities

Capital activities are dependent on the nature of the target's long-term capital management. As inputs, activities, and outputs describe the operations of the target, capital activities describe the assets.

Capital appreciation: This is rare outside of targets in finance or real estate. Technology targets may list patents and IP here. Otherwise this is for assets that appreciate in value.

Capital depreciation: Depreciating assets and amortization of investments (if not listed in capital investments).

Capital investments: One-time spend on asset acquisition or maintenance, either from the target's internal capital accounts or from external investors.

Capital expenditures: Ongoing capital expenditures not related to specific inputs, activities or outputs. This includes cost of sales, general expenditures, administrative fees, legal fees, marketing (sometimes called out separately), and any other ongoing expenditures.

This also includes cost of capital, if outside financing is raised, and interest on capital.

Net

Total Capital Expenditure (or Appreciation) = (Capital Appreciation)-(Capital Depreciation)-(Capital Investments)-(Capital Expenditures)

Pre-Tax Income (PTI) = (GP)-(Total Capital Expenditure)

PTI% = (PTI)/(Revenue)

Executive Summary

Pull the important lines from each target's business case into a master executive summary. This should include the same parts described above, and should show the reader how each target impacts the overall business case.

Examples

Templates

Document	Why its good	Key lessons
https://r3share.mohso.com/dl/ACdLYZGfqv	Excellent example of Business Case to support DLT in Industry	Engage in project where R3 gained
https://r3share.mohso.com/dl/YDm1Or03at	Excellent example of Business Case to support Cash on DLT	Engage in project where R3 gained

Product Design and Build

Overview

This section describes Product Build and Design for projects that include building a CorDapp.

Principles and good practices

Principle 1 - Determine the Project Sponsors

Welcome to Corda Solutions!

Define the accountability structure among core & participating members – every project must have strong sponsorship:

The sponsor can be a member/groups of members or a partner/group of partners. The sponsor needs to be committed to making the project happen and to funding it.

Examples:

- Member sponsor: Wells Fargo for Janus
- Partner sponsor: Calypso
- SI/ISV sponsor for Jasper III
- Joint Venture (HQLAX)

Considerations for project accountability structure:

- Identify the Product Owner who will have accountability for product-specific decisions
- Identify the stakeholders and agree a steering committee
- Define Team Structure: roles & responsibilities based on background and subject matter expertise
- Consider legal and regulatory issues (link to Legal and Regulatory Stream)
- Consider questions of IP ownership structure (link to Legal & IP Stream)
- Build agreement on outcome: the vision for the product and whether prototype, pilot, or full production CorDapp

Principle 2 – Role of Product Owner

Assign the Product Owner & Document the Product Vision

The Product Owner (not necessarily the sponsor):

- Clearly states the business case and champions the vision and the outcome of the Product: what will the future state look like? (link to Business Case stream)
- Drives the product strategy: how to get there
- Ensures appropriate support, access to resources, communication and successful overall results
- Manages the roadmap, and the stakeholders
- Work with the team define feature requirements and groom and manage the backlog
- Estimates costs of build and adoption, and savings projections

The team needs to both acknowledge the leadership role of the Product Owner and expect strong direction on the product build.

Principle 3 – Customer Driven Design

Determine the Customer needs & document the Requirements

The product owner and team will:

- Identify pain points and possibilities for improved workflows and technical functions
- Re-engineer current business models to transition to a distributed model - map to the future state with a DLT infrastructure
- Create user stories, story boards, mock-ups as needed and include in backlog
- Define feature and functionality requirements
- Define technical (non-functional) requirements
- Document acceptance criteria and define ‘done’

Principle 4 Architecture Design

Describe the Software & business architecture, including Business Network Software Architecture

- Define MVP solution based on the prescribed DLT business model
- Production considerations e.g. Technical Pilot setup, Business Networks, Privacy rules, Default protocols, Application Governance & System Integration
- Define Success Criteria
- Ensure that the design is capable of meeting evolving needs – bug fixing, upgrades, new features.

Business Architecture

- Further validation of the business case
- Business Model and Go-To-Market Effort (link to Commercial Model Stream)
- Map Process flows - existing & new
- Map transaction lifecycle
- Define all events, actors
- Interview potential partners

Business Network:

- Governance and operational framework for the use case
- Initial legal/regulatory basis for the BN
- Define Notary, Doorman and Operator
- Align with evolving BNO Playbook

Principle 5 - Validate with Customers

- Involvement of all participants in definition, design, and prototyping sprints and retrospectives
- Agreement and planning of testing and acceptance criteria
- Continuously inspect and adapt processes
- Product validation: SMEs external to project to review product feasibility, product design, and to participate in User Acceptance testing

Principle 6 - Platform Requirements

Communicate Corda platform requirements and CorDapp design to Product team

- Assumption: the partner or member building the CorDapp has already been trained on Corda
- Provide requirements, from the team building the CorDapp, to the Corda Platform team, e.g. firewall float
- Review Cordapp progress against Corda roadmap with Solution Architect to ensure alignment with Platform team

Principle 7 - Integration Architecture

- To deploy DLT solutions into the financial institutions:
- Provide enough data and information to allow Compliance to accept Corda
- Provide enough data and information about the CorDapp
- Perform gap analysis on all processes, functions, roles, etc that are being re-created in DLT, and those that are not and map existing dependent business processes to DLT infrastructure & new product
- Estimate costs and savings implications

[R3 INTERNAL] Legal

Overview

This section discusses how to build the legal architecture for your CorDApp solution.

Based on the experience of numerous previous projects these are considered pre-requisites:

- Establish proposed product.
 - Establish business case.
 - Establish commercial model.
 - Establish operating model.
 - Establish proven technology solution.
 - Test (i.e. gain feedback on) all of the above with market participants i.e. users who ultimately will help generate revenue for the platform (or achieve alternative objectives).

Principles and good practices

Principle / Practice 1

Establish clear business case and justification for legal expense

- This is hopefully an output from (1).
 - Be clear on why you need to hire a legal provider e.g. to support tokenization of assets on DLT.
 - Be clear on why the expense needs to be incurred now rather than later e.g. “it’s needed now to ensure participants can use the intended platform in 12 months time”.

Principle / Practice 2

Obtain funding solution for legal expense

- Establish who will pay the bills for the legal provider.
 - Example funding solutions include:
 - Cash investment from one or more stakeholder (e.g. consortium members or working group or single product company or technology provider).
 - Legal provider expenses paid in the form of equity of a company (e.g. CorDapp builder).
 - Note: there is usually correlation between the IP owner and the funding solution.
 - Establish the minimum and maximum amount you are prepared to pay for the services described in the RFP.
 - Confirm whether additional taxes (like UK VAT) will be payable e.g. based on jurisdiction of the paying entity.

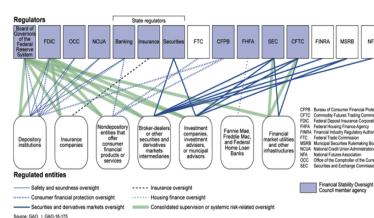
Example of Comparison Matrix

Principle / Practice 3

Create RFP

- There are plenty of tips / guides / training materials out there on how best to write an RFP, here are a few:
 - <https://www.grfcpa.com/resources/articles/how-to-create-an-effective-rfp>.
 - <https://www.wikihow.com/Write-a-Proposal>.
 - <https://www.banyanrfp.com/2013/06/structuring-an-rfp-for-legal-services-the-basics>.

- Be as clear as possible e.g. describe measures-of-success or artifacts that you expect to exist at the end of the engagement.
- Ensure deliverables deal with:
 - i. Key elements of the product operating model e.g. token feasibility, classification of asset tokens, who issues the token, legal enforceability of tokens, fungibility of tokens, access to, perfection and enforcement against the assets, privacy rights of participants.
 - ii. Any settlement finality required with respect to assets changing ownership via DLT.
 - iii. Regulatory licensing requirements e.g. for financial services (if applicable) MIFID reporting, CSDR, MTF, anti-trust rules etc... that are relevant to the product being built (see separate section on Regulatory).
 - iv. Jurisdictions – which jurisdictions should be considered in scope for the legal architecture that you intend to be built.
 - v. Documentation that you think will be needed to support the intended operating model of the CorDapp e.g. platform rulebook.
 - vi. Timing / Phasing requirements – when certain documentation is needed by e.g. for a pilot transaction or a platform launch.
 - vii. Any upfront guidance on liability provisions / levels, data usage requirements.
 - viii Clarity on signatures i.e. who needs to sign which documents (see table below).
 - .
- Provide any over-riding guidance or principles e.g. “the pilot documentation should be done in a way that’s consistent with the end-state platform legal architecture”, or “existing GMSLA documentation between participants should be unchanged”.
- Be clear on whether you expect the legal provider to be part of the legal negotiation process required to get the end-product signed by all parties to the legal documentation (if the answer is no, then be clear on who will play this role).
- Be clear on how the legal provider will be paid for their services (pre previous section).



Principle / Practice 4

Create shortlist of legal providers for RFPs (request for proposal)

- Consider past experience and existing relationships from working with legal providers.
- Consider market reputation of legal providers.
- Consider recommendations.
- Consider expertise levels and how well they match the operating model of your CorDapp e.g. knowledge of similar DLT solutions, of relevant knowledge of asset ownership transfer solutions on DLT.
- Consider complexity / depth of product expertise required to build the architecture to support the CorDapp and operating model.
- Consider jurisdictional expertise of legal provider for the CorDapp product e.g. you may want to build for one region first, so which legal providers have expertise in that region?
- Agree a suitable size of shortlist e.g. 5 providers.

Example Excerpt from Potential Provider

Norton Rose Fulbright is one of the world's leading finance and FinTech law firms, with long-established practices in the UK, Netherlands and France and in 50+ jurisdictions around the globe.

Our French practice is listed in the global top 10 (Chambers 2018) and takes a broader industry role (e.g. co-chair of the fintech alliance) in addition to regular advisory mandates.

As a firm:

• In-depth, correct knowledge – e.g. please see the short notes later on page 11 (French law – recent position relating to

DLT in the French market) and the attached publication draft of the latest article by the NRP Paris team on tokenisation, to be published shortly in the French Bulletin July/Bourse; and

• quick, clear and practical advice – e.g. please find together with this proposal an initial track up by Vély Jones of the DLT-based ECP environment, and the key issues to consider and potential role with the brought up and (if any) changes may be needed in a DLT in a traditional ECP environment.

Please see Appendix 2 for sample credentials.

Top 5 Global Elite Law Firm

(Acritics 2017)

"a noted thought leader in the blockchain and smart contracts space"

(Chambers FinTech 2018)

Finance Team of the Year -

Best Commercial Finance

Impact Fund

(Legal Business Awards 2018)

Principle / Practice 5

Engage legal providers

- Send RFP to shortlist of legal providers.
- Try to meet legal provider team face-to-face e.g. via multi-hour workshop to explain the goals of the product, the operating model, the RFP and what is expected of them.
- Clarify / answer any questions legal providers may have about scope.

Principle / Practice 6

Negotiate

- Negotiate a better price / discount.
- Emphasise value of any potential follow-on work.

Principle / Practice 7

Select legal provider

- Clearly determine stakeholder or group of stakeholders that need to make decision.
- Ensure all stakeholders have a voice and are part of decision making process.
- Factors to consider:
 - Price.
 - Any upfront discounts (in view of follow-up work).
 - Timeline.
 - Reputation / experience.
 - Detail of quote i.e. how much effort they put into the quote, and their understanding of the piece of work.
 - Desire / willingness of legal provider to do the work.
- Use a score sheet on the above points to record the reasons behind the collective decision.

Principle / Practice 8

Assign project manager to oversee engagement with legal provider

- Don't rely on legal provider to do this role, because the project manager is in effect ensuring the deliverables are met on behalf of the stakeholders (who are paying for the legal provider services).
- Ideally establish single point of contact with the legal provider – often easiest if this person is also the project manager.
- Role of project manager:
 - Set priorities with law firm on behalf of client.
 - Manage / communicate change to deliverables.
 - Provide status to stakeholders.
 - Ensure value for money is being achieved for the client.

- Establish communication strategy with legal provider e.g. weekly calls, additional workshops etc....
 - Ensure deliverables met on time.
 - Agree invoice schedule and manage budget with the legal provider.
- Agree more detailed set of deliverables with legal provider early on in the engagement and use that to provide and communicate status during the engagement. For example, use a tracking table like the one below which summarises:
- All legal documents that need to be built.
 - Who needs to sign each one.
 - Status of each document being built - from a readiness perspective, negotiation perspective, and signature perspective.
 - If multiple legal providers are required, then add a column to clearly state (for benefit of all parties involved) which legal provider is building each document – establishing clear accountability and ownership.
- Ensure legal documents are signed by the right people within each organization (sometimes the wrong person initially signs within an organization, which can delay things) - ask each party to provide list of authorized signatures for each document.

Platform Agreement / Signatory	Responsibility for legal build	Participant 1	Participant 2	Custodian	CorDapp Owner	R3 Limited
GMSA	Participants - already exists	SIGNED	SIGNED	Not Required	Not Required	Not Required
Custody Agreement	Custodian - already exists	SIGNED	SIGNED	SIGNED	Not Required	Not Required
Platform Agreement	Legal Provider	SIGNED	SIGNED	Not Required	SIGNED	Not Required
CorDapp Software License Agreement - Custodian Version	Legal Provider	Not Required	Not Required	SIGNED	SIGNED	Not Required
CorDapp Software License Agreement - (Participant Version)	Legal Provider	SIGNED	SIGNED	Not Required	SIGNED	Not Required
R3 Net Agreement	R3 Legal	Not Required	Not Required	Not Required	SIGNED	SIGNED
R3 Net Participant Terms of Use	R3 Legal	SIGNED	SIGNED	SIGNED	SIGNED	Not Required
Control Agreement	Legal Provider	SIGNED	SIGNED	SIGNED	Not Required	Not Required
Control Agreement	Legal Provider	SIGNED	SIGNED	SIGNED	Not Required	Not Required

Principle / Practice 9

Negotiation of legal provisions with entities that will sign the legal documents

- This can sometimes be a protracted phase of the project, involving lengthy backwards and forwards between parties regarding the detail of the provisions laid out in the legal agreements.
- Ensure everyone understands who is overseeing this phase and acting as the conduit between negotiating parties - often this role is played by the project manager, but that doesn't have to be the case.

Principle / Practice 10

Wrap-up workshop

- Review all deliverables and agree which ones have been delivered vs. deferred vs. any changes in scope along the way.
 - This may be a requirement for releasing final payment funds to the legal provider.
- Review lessons learnt along the way.
- Review any next steps or further potential pieces of work for the legal provider.

Examples

Document	Why its good	Key lessons	IP / Distribution Rights
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal

[R3 INTERNAL] Legal and IP Framework

Overview

The purpose of this section of the framework is to understand how R3 deals with Legal and Intellectual Property Agreements that are put in place with Partners, SI's and Members

In addition we will review the R3 Legal Document Suite, and the typical use cases for which each artifact is most appropriate.

List of Legal/IP Artifacts

- Non Disclosure Agreement
- Professional Services Agreement
- Statement of Work
- R3 Corda License Agreement
- R3 Corda Evaluation License Agreement
- Code of Business Ethics (potentially sent by Partner, SI, ISV as requirement for engagements)

History

Projects executed prior to the establishment of Global Collaborative Laboratory were done under the Advisory Services Agreement.

Once GLC was created all subsequent engagements were undertaken with members signing a Master Services Agreement (MSA).

There were some select projects that involved the participation of technology vendors e.g. Axoni which used Technology Provider Agreement (TPA).

Upon the release of Corda in November 2016 the landscape changed, GLC went away and we now view legal agreements thru the lens of Corda Open Source vs Corda Enterprise.

The different agreements needed depending on which version of Corda is used will be discussed in more detail below.

Principles and Good Practices

When determining the type of agreements that should be entered into there are a few things to take into consideration.

Principle 1 - Determine Type of Engagement

The first prerequisite in determining the appropriate agreement is the type of customer engagement:

- Partner
- SI/ISV
- Joint Venture
- Bank Lead

The relationship R3 has with parties to a legal agreement will determine the nature and type of agreement:

- Partner developing their own product
- Member developing IP within a POC
- Systems Integrator developing IP on behalf of their client
- Determine what constitutes contributed IP
- Determine what constitutes developed IP
- Determine ownership of contributed and developed IP

Principle 2 - Determine Services Supplied

The next prerequisite is determining what R3 will be supplying in the engagement, as each of these may require separate agreements.

- Software
- Services
- Research
- Training

Principle 3 Determine Key Parties

At this point it is worth pointing out, though this may seem obvious, who are the key parties responsible for contract negotiation:

- R3 Counsel
- Client Counsel
- Partner Counsel

While the Services team are active in ensuring there is ongoing communication between the legal teams, final sign off must be completed by our attorneys. If there is a business decision needed to finalize a legal agreement then that can be channelled appropriately to a member of the R3 Management Team.

Principle 4 - Determine Corda Version

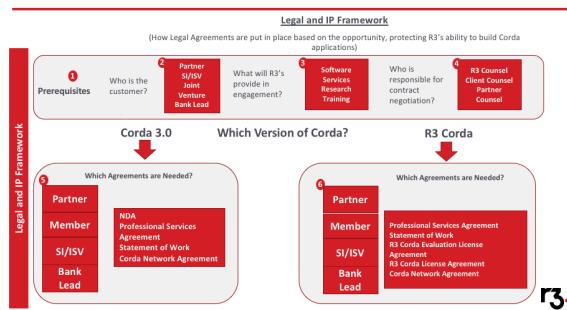
Corda Open Source vs Corda Enterprise

Corda Open Source engagements typically require the following agreements in place:

- Non Disclosure Agreement
- Professional Services Agreement
- Statement of Work

Corda Enterprise engagements typically require the following agreements in place

- Professional Services Agreement
- Statement of Work
- R3 Corda Evaluation License Agreement
- R3 Corda License Agreement



Principle 5 - Determine IP Requirements

Here we will discuss the types of IP involved and the different considerations given to each. At the outset we should establish basic goals from an R3 perspective. Fundamentally we want to:

1. Ensure R3 has rights to IP from all projects where possible to enhance the Corda Platform
2. Ensure R3 has the ability to develop commercial products on Corda platform which may be similar to those where R3 has no rights to IP.
3. Ensure material liability is at a minimum.

Pre Existing or Contributed IP

It is quite common for a Partner/Vendor will bring in what is known as contributed or “pre-existing IP” i.e. technology components that existed before Corda and which are not owned by R3. R3 does not usually seek any rights/claims to pre-existing IP.

An example would be an FX matching algorithm developed by a Partner which uses Corda to produce a ‘golden copy’ of a matched FX trade. R3 did not develop this functionality prior to the engagement hence has no rights to the IP for that functionality.

Developed IP

R3’s fundamental approach to IP agreements is to have an unrestricted license to all developed IP during the course of an engagement.

The reason for this is simple, R3 wants the right to be able to take what is learned in every engagement and determine if a platform component should be added as a result of that learning.

For example, if a specific RDBMS is used for the Corda Vault during the course of an engagement, and changes are needed to Corda to facilitate the use of that RDBMS, then R3 will want the ability to port those changes to the platform so they become an integral part of the product.

Non Exclusivity

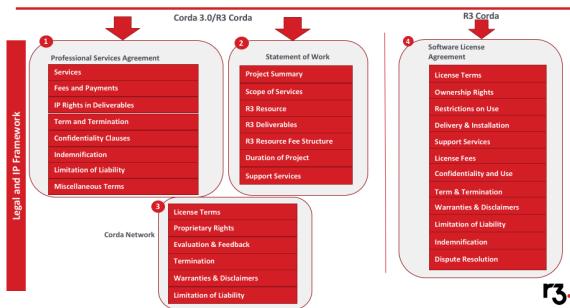
One of the most crucial clauses in any R3 IP agreement is that of Non Exclusivity. After the sections around IP ownership, this is arguably the next most important clause to take into consideration.

This clause has one major purpose, which is to safeguard R3’s right to engage in similar business, opportunities etc whether or not they be similar to or competitive with an existing R3 engagement.

It is intended a form of protection against a case of IP infringement being brought against the company. This is a sample extract of the clause:

R3 may carry on existing businesses, seek or make other business opportunities or investments, enter into new lines of business and/or develop or market new or existing products or services in any jurisdiction or territory or with any third parties whether or not the same as or similar to or competitive with any Services, Deliverables or any other services provided or deliverables developed pursuant to this Agreement, provided always that in doing so it does not breach any of its obligations hereunder.

General Terminology



Terms Specific to R3 Corda Software License Agreement

License Grant

R3 grants Licensee a worldwide non-sub-licenseable,non-transferable license to R3 Corda. Licensees have right to develop API's & Linked Applications

Ownership Rights

Licensor owns all right, title and interest, including IP to R3 Corda Software. Licensees agree not to alter proprietary markings. No reverse engineering of Corda is permitted.

Restrictions on Use

Licensee agrees to avoid unauthorized access to Corda Software and prevent unlawful distribution.

Confidentiality and Use

Licensee agrees not to disclose any confidential information between parties including all non-public records, books, contracts, reports, instruments, computer data and other data. In general the only time confidential information can be divulged is when parties are compelled to do so by laws.

Terms and Termination

Provides details on the length of the agreement and the reasons for possible termination of license by R3.

Templates

The following are the most recent documents available from R3 Legal.

Document	Purpose	Owner	Last Updated
https://r3share.mohso.com/dl/9SrRcFLbYb	Non Disclosure Agreement	R3 Legal	Feb 2018
https://r3share.mohso.com/dl/5hXWTZI0FN	Professional Services Agreement	R3 Legal	Mar 2018
https://r3share.mohso.com/dl/2Mno8Q9VRU	Statement of Work	R3 Legal	Apr 2018
https://r3share.mohso.com/dl/oEZEEn8KVbe	Enterprise License Agreement	R3 Legal	Mar 2018
https://r3share.mohso.com/dl/fe6iazT57n	Enterprise Evaluation Agreement	R3 Legal	Feb 2018

Examples

Document	Why its good	Key lessons	IP / Distribution Rights
https://r3share.mohso.com/dl/09ryJFddnZ	Excellent example of Services SOW to support SI Partners	Engage in project where R3 gained little in terms of IP or \$\$	Non R3, Client Owned
https://r3share.mohso.com/dl/btQs6umnQx	Details IP ownership and Non Exclusivity	Engage in project where R3 gained little in terms of IP or \$\$	Non R3, Client Owned

[R3 INTERNAL] Regulatory

Overview

How to build the regulatory approval framework for your CorDapp solution.

This are the pre-requisites:

- Establish proposed product
- Establish business case
- Establish commercial model
- Establish operating model
- Establish proven technology solution
- Proof test (i.e. gain feedback on) all of the above with market participants i.e. users who ultimately will help generate revenue for the platform (or achieve alternative objectives)



Principles and good practices

Principle / Practice 1

Perform a regulatory risk assessment for your CorDapp

- Perform a risk assessment to determine higher / lower regulatory touchpoints
 - For example, use the risk assessment scoresheet below (developed by R3)
 - High risk projects are generally more likely to require a regulatory strategy
 - Some CorDapp owners have also used a third-party risk assessment provider like www.riskbusiness.com to assess the overall operational risk benefits to clients using the CorDapp
- Characteristics to consider:
 - Types of clients (retail vs. institution)
 - Complexity vs. simple product
 - Whether any licenses are required for the CorDapp solution
 - Competition
 - Which jurisdictions and countries are in scope for the platform
 - Whether tokenized assets need to move across jurisdictions
 - Which existing regulated products are involved in the CorDapp operating model e.g. existing master agreements
 - Which associated governing bodies help to govern these existing products e.g. ISMA, ICMA
 - Whether the operating model will involve central banks in any way e.g. CBDC or to help pledge assets to a central bank
 - Legal characterization – of key parts of the operating model e.g. tokens
 - Qualification of the system - trading venue, clearing and/or settlement system, or none of the above
 - Regulatory governance of the business network for your CorDapp
 - Understanding what AML/KYC and other financial crime obligations apply
 - Concurrent regulatory changes e.g. GDPR, Brexit
 - Cybersecurity / data privacy / record retention: i.e. what information will be recorded on the blockchain
 - Any regulations that may prevent the solution from going live

Regulation / Interpretation / Procedure	Themes	Low	Moderate	High	Rationale
Technical	Regulatory Reporting	The product / service output will not be used to file regulatory reports, or the product / service does not respond to regulatory requests.	The product / service output is used to file regulatory reports, or the product / service does respond to regulatory requests.	The product / service output are used to file regulatory reports, or the product / service does respond to regulatory requests.	Information from the DLT is used to file regulatory reports, data integrity needs to be maintained, and the DLT needs to be engaged with third parties. Regulation must be engaged.
Commercial	Regulatory Proofing	The intent of the product / service is not to adduce a present regulatory need.	The intent of the product / service is to adduce a present regulatory need.	The intent of the product / service is to adduce a present regulatory need, but can keep client's compliance concerns.	The intent of the product / service is to adduce a present regulatory need, but can keep client's compliance concerns, and may be concerned with third parties. Regulation must be engaged.
Functional	Privacy and Confidentiality	R3 will not be able to provide information about the client or their data to R3 through the network.	The nature of the product / service requires R3 to have access to client information, which will be shared with the network.	The nature of the product / service requires R3 to have access to client information, which will be shared with the network.	The nature of the product / service requires R3 to have access to client information, which will be shared with the network. This will be done in accordance with the DLT's terms of service.
Operational	Reputation	Product / service is simple and issues will not lead to a notable disruption to the network.	Product / service is moderately complex and may issue, which can result in significant loss, disruption or damage to the network.	Product / service is extremely complex and may issue, which can result in significant loss, disruption or damage to the network.	Reputation risk often has negative public control. This can negatively impact the DLT's reputation, and may result in significant loss, disruption or damage to the network.
Commercial	Fines / Penalties / Regulatory Issues	The product / service will have no impact on existing regulatory fines / penalties.	No fines or known penalties / fees, but design can help client's address regulatory issues.	Massive fines / penalties / regulatory issues related to the usage.	If the product / service helps mitigate regulatory fines / penalties, client will be less likely to face more significant costs of regulation.

Principle / Practice 2

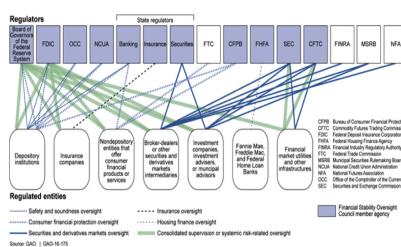
What - identify any regulations that your CorDapp will help participants meet

- This will likely be part of the business case for building the CorDapp e.g. “a collateral token application may help participants meet their LCR regulatory requirements, as well as various collateral margin regulations related to OTC derivatives, and therefore reduce capital costs for participants of the platform”
- Is the CorDapp aiming to disrupt an existing marketplace which itself is subject to regulatory oversight?
- How will regulatory integration benefit users of the platform e.g.
 - Reduce operational cost
 - Reduce regulatory fines
 - Reduce technology costs
 - Reduce regulatory reporting costs
 - Reduce operational / technology risk
 - Increase regulatory transparency e.g. is there value in CorDapp having a regulatory observation node to observe transactions on a real-time basis?
- Will the CorDapp will be used to produce regulatory reports on behalf of clients - if information from the DLT is used to file regulatory reports, data integrity audits will need to be conducted by a third party, and regulators will likely need to be engaged
- Is the intent of the CorDapp is to address a present regulatory need or concern - The ability to address a regulatory concern, intentionally or unintentionally, can peak the interest of regulators and compliance
- Are there massive fines / penalties / regulatory issues related to the intended market place of the CorDapp - if the product / service helps mitigate regulatory fines / penalties, then regulators should also be consulted
- Review these regulatory benefits (along with the business case of course) with potential participants to take onboard their feedback as early as possible

Principle / Practice 3

What - identify oversight regulators for your CorDapp solution

- Identify regulatory agencies that will oversee this solution (if any):
 - Local and global regulation(s)
 - Example regulations include: GDPR, SFTR, CSDR, Consumer regulations (TILA, UDAAP), exchange regulations (e.g. MTF etc...), MIFID II, LCR, BIS, BASEL III, Dodd-Frank, PFMI, IFRS9
 - Add list of regulators here from Neepa?
- R3 regulatory affairs has excellent expertise in this area and can help
 - R3 engages with over 100 national and international regulatory bodies and central banks around the globe, within the various jurisdictions in which R3 members are domiciled, to discuss the impact DLT will have on, and how DLT can be leveraged to support, regulators' current and future needs
 - For example, the diagram below gives an overview of product regulators in the U.S.



Principle / Practice 4

- Establish who involved in the operating model needs to adhere to which regulations:
 - CorDapp builder / owner

- Users of the CorDapp solution e.g. LCR for banks / clients
- Third-party / infrastructure providers e.g. CSDR for custodians
- Commercial banks, technology providers, central banks
- Technology solution providers

Principle / Practice 5

Based on the impacted regulations, determine if regulatory functionality should be embedded in the CorDapp technical design, operating model, or if a new business case should be developed

- If yes, then revise technical design / operating model or business case

Principle / Practice 6

- Form your strategy based on the conclusions of the previous steps
- Determine who you should form your strategy with:
 - Just yourself i.e. CorDapp owner / builder?
 - With client regulatory affairs teams?
 - With any infrastructure providers or third-parties who are helping to build the CorDapp solution?
 - With R3 Legal and Regulatory Affairs Team to help define the regulatory outreach strategy?
 - Consider whether it makes sense to establish a regulatory outreach working group with any of the above groups
 - Determine whether a legal provider can also help aspects of the regulatory outreach strategy
- Determine list of regulators to communicate with, and measures-of-success for each regulator (examples include):
 - Prioritise by intended roll-out by jurisdiction
 - Seek feedback from regulators
 - Review, whilst avoiding any negative feedback
 - Will regulators provide approval? Generally regulators unlikely to approve a product because they don't wish to be seen as favouring any particular market provider
- Determine communication strategy:
 - How to communicate with regulators e.g. via customers, directly, or otherwise
 - When to communicate with regulators
 - Early on to help shape the solution design?
 - A bit later after the solution has been proven e.g. via a first live transaction, or later still when there's something closer to production to show them?
 - How often to communicate with regulators? Some more frequently than others?
 - Involve regulators directly in project(s) / tests / pilots / live transactions along the way?
 - Determine in advance what regulator will want to know about a new 'blockchain' solution? e.g.
 - Benefits of doing this on a blockchain vs normal means?
 - What happens in the event of a defaulting participant who can no longer honour their obligations?
 - How safe are any assets represented on blockchain?
- Fully document your regulatory outreach strategy so that everyone involved is aware of roles and responsibilities
- R3 regulatory affairs has excellent expertise in this area and can help

Principle / Practice 7

Execute your regulatory outreach strategy

- Record / document all meetings with regulators, including agreed follow-ups and timing of follow-up meetings
- Based on any feedback from regulators, be prepared to modify specifications if it makes sense to do so

Principle / Practice 8

Continue to re-assess and execute your regulatory outreach strategy after CorDapp launch

- Repeat earlier steps if required

Examples

Document	Why its good	Key lessons	IP / Distribution Rights
https://r3share.mohso.com/dl/KI8SliYKh	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal

Go To Market

Overview

What to think about:

- What are you selling?
- Who are you selling it to?
- How will you reach your target market?
- Where will you promote your CordApp?

This is an interactive process which will continue to evolve as you go through the project lifecycle.

Principles and good practices

Principle / Practice 1

What are you selling?

- Use customer validation

This is about exploring issues, understanding phenomena, and answering questions by analyzing and making sense of unstructured data. For previous projects like Marco Polo we used various Qualitative Research methods throughout the process to validate the Idea, Value Proposition, Concept and the various Prototypes. For more info and templates:

<https://r3-cev.atlassian.net/wiki/spaces/CLWG/pages/132164157/Corporate+validation>

- Include ecosystem early on in the creation of the whole solution. In a distributed ledger product world we need to think of all the components that will play a role in delivering the whole solution - e.g.
 - Hardware
 - Software
 - Billing system
 - Service support

- Installation
- Cloud based UI
- Data analytics/Data providers
- Value added services/Oracles
- *Tie this to the Partner strategy work*
- What is the competitive elements of the Business Network? what is DLT providing to either an existing product improving the current offering or what new business models are being created?
 - Pain Points => implications for customer (e.g. Bank) => Implication for customers customers (e.g. Corporate) => Opportunity fit
- Commercial/Pricing - work from the start with the commercial and sales team to ensure different commercial arrangements and considerations are being made
 - Transaction based licensing vs entity based licensing
 - On-prem vs Cloud deployment
 - Architectural considerations (e.g. how many nodes needs to run, how often they need to be active, what notary services does the solution need, representative nodes, 3rd management and hosting etc.)

Principle / Practice 2

Who are you selling it to?

Who is your segment?

- Define your customer market
- If relevant define your customers customer - this is required to ensure the product fit and suc Any bugs or requests for new features that arise because of a support requests will be recorded within the ticketing and support system, and updates will be provided to the customer outlining which release the bug/new feature will be included in.

Principle / Practice N

How to Get support

There are several primary ways to get support during the Marco Polo phase.

1. Consult the TradeIX customer Knowledge base: [click here](<https://support.tradeix.com/hc/en-us>)
2. Post questions in the TradeIX customer community forums: [click here](<https://support.tradeix.com/hc/en-us/community>)
3. Raise a ticket via the "submit a request" button on the customer knowledge base website.
4. Send an email to support@tradeix.com

SLA

Below are the outlined estimated SLA response and solution times

Templates

Include any document templates here. Link to the source file in the master location. NO POWERPOINT, ever.

Document	Purpose	Owner	Last Updated
Filename / link here	Short text description	Owner name here	Date here
Filename / link here	Short text description	Owner name here	Date here
Filename / link here	Short text description	Owner name here	Date here

Examples

Include any client examples here to make it more real. Any examples should be uploaded and linked to in PDF format – always mark files with the IP licence attached (i.e. r3 internal, r3 consortium, public).

Document	Why its good	Key lessons	IP / Distribution Rights
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal
Filename / link here	Describe why this is a good example	Key things you would not do again!	public / r3 consortium / r3 internal

Corda Business Networks

Introduction

In 2016, the design concepts for Corda were described for the first time in Mike Hearn's [technical whitepaper](#). It was to be decentralised global database, providing a platform for decentralised application development. The paper described the concept of "CorDapps" (Corda Distributed Applications) and how these might be constructed, but did not attempt to describe very much more detailed thinking. Building and managing distributed applications is, however, more involved than just creating software: In a CorDapp, information and/or assets are exchanged, or transacted, between participants because they represent something meaningful to each of them, but how are those meanings determined, on what basis, and by whom? Such questions are the ones we seek to consider and understand.

Superficially, our problem is not particularly new. There are many examples, predating any sort of blockchain or DLT (Distributed Ledger) technology, of companies and consortiums that have brought people together to transact value. The blockchain vision enabled by Corda is different, however, because it was designed to eliminate many of the problems inherent in earlier approaches.

First, let us give our problem a name: We have a group, or network, of independent parties that want to use Corda to transact something of value to each, and as we're transacting something valuable then this almost inevitably implies a commercial, or business, purpose. As such, we can talk about a "Business Network".

What is a Business Network?

If a business network is a group of independent parties transacting together, then its purpose is to allow its members to create a shared representation of information, or facts, and to then use shared processing of those facts to achieve agreement, or consensus, about operations involving them.

This ability to enable both shared understanding of facts, and shared understanding about how they are to be used is something uniquely powerful within DLT/blockchain systems. Earlier systems focused on the shared representations of information, but could neither consistently guarantee its correctness, nor ensure that all participants processed things in the same way. The Corda promise is that "I know I see what you see" after each operation to which we are both a participant.

The use of shared business logic is necessary to ensure correctness, but also offers a new opportunity. Where our transacting parties are all required to perform similar business processes then this can be designed into the CorDapp(s) that are developed. This means that when it comes to integrate Corda business networks into new, or existing, systems, that total integration effort is notably lowered, as each participant is not left duplicating the same basic functionality.

The model of Corda business networks also enables something particularly powerful. It allows for the possibility that one business network can build upon the work of another, and that others can then build on top of that.

What are the Challenges Associated with Business Networks?

Corda enables business networks, but deliberately sets out to have few "opinions" about what a business network might be, or exactly how it should work. Instead, Corda attempts to define some mechanisms to allow for the

construction of business networks, and leaves the rest rather open-ended. This flexibility means it is possible to build both very simple and very complex designs, but, as with most software, over-simplified designs often miss essential functionality, while over-complex ones are almost impossible to get right.

How Do Business Networks Start?

Now we have an idea of what a business network is, we might now ask how a Corda business network gets started?

There is no single approach that describes this, but there are a few common routes. Sometimes a business network already exists, and it simply adopts Corda, while in other cases a governing consortium might be pulled together first. Another route is that software developers find an opportunity to build out a business network from a prototype or proof-of-concept design. It is worth noting that while there are many possible journeys to creating a Corda business network, the outcome is, generally, similar.

Top-Down (Business-Led)

A top-down journey usually starts by recognising a valuable business problem, identifying likely participants, and, perhaps, the creation of a new venture, or consortium. Typically, this route requires the creation and adoption of technical standards, along with definition of governance principles. This usually evolves to include specialised resources such as Corda developers and legal advisors.

It is possible that a suitable structure already exists, so in these cases our goal is to adopt, or evolve, its business model, and to embrace as many of its existing governing policies and procedures as make sense.

Bottom-Up (Technology-Led)

Sometimes a business network comes about as the result of a developer or technology team building a prototype, or proof-of-concept, of some interesting technology idea that can be applied to a business problem. In these cases, the technology has to be matched to a suitable problem and governance structure. While it may be easy to dismiss this as a solution looking for a problem, there has been a surprising degree of success with this type of approach, and this embraces the possibility of very simple solutions to non-controversial problems.

Roles and Responsibilities

We have considered how Corda business networks are started and have seen that the process leads to some sort of governing entity. That governing entity addresses the strategic business and technical needs of the users of the business network's services, but does not address the tactical, operational, needs. Similarly, our users have not really been considered at this point, so it is worth exploring these key roles.

It is worth noting that these roles do not come in one size and may, for example, represent an individual, a company, or a consortium.

Business Network Governor (BNG)

The Business Network Governor (BNG) is responsible for the strategic management and governance policies of the business network, and is accountable for the implementation and enforcement of those policies.

- Defines the legal and regulatory basis on which the business network operates.
- Establishes and controls any technical standards associated with the business network's activities.
- Sources and maintains the CorDapp (and other) software necessary to meet the needs of the business network.
- Determines how the business network will be operated and supported.

It is sometimes argued that if we were to build a truly decentralised business network then there is not role for a BNG, but even the decision to make the network decentralised implies a specific governance decision. The BNG may not be readily visible, but does not go away.

Business Network Operator (BNO)

The Business Network Operator (BNO) is responsible for the tactical day to day management of the business network.

- Implements and enforces the policies established by the BNG.
- Manages membership of the business network.
- Handles marketing, promotion, and any other day to day organisational services, that are required to operate a healthy business network.
- Provides, or arranges for, support for problems encountered by the business network's users.
- Provides reporting to the BNG.

It is worth noting that for some business networks the BNG and BNO role may be fulfilled by the same organisation.

Business Network User (BNU)

A business network user makes use of the services offered by the business network. There may be many different types of user, depending on the application.

Successful Business Network Deployment (The Playbook)

Having set the scene for Corda business networks, it is useful to think about some specific tasks and their implications for how any particular business network will operate. We introduce some thoughts in the form of a “playbook”.

The playbook is not intended to be exhaustive and is a living document. More content will be added as everyone gains insight into what makes for the best practices for all participants within a Corda business network ecosystem.

S. Structure

Our business network requires some sort of operating structure. This may mean we are reusing, extending, or repurposing, an existing organisation. We may also be creating a new entity.

S.1. Define Business Network Mission

- Describe the purpose of the network. If existing, does repurposing the network on Corda alter the mission?
- Describe asset type(s) involved, exchanged, affected in the business transactions.
- Describe the participants at a high level (details covered later).

S.2. Assign Business Network Roles

- Business Network Governor (BNG)?
- Business Network Operator (BNO)?
- If the BN already exists, are there roles already established that should be included or mapped to the above roles

S.3. Define the Governance Body Model

- Model structure, e.g., hierarchical, committee, etc.
- Responsibilities
- References:
 - <https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Financial-Services/dttl-fsi-US-FSI-Developing-a-Governance-Model-for-Business-Networks.pdf>
 - <https://pwcseminars.citibank.com/275ad750e55f02241bb6f7c1c36c586d86d75fc3b.pdf>

G. Governance

G.1. Describe Participants

- What are the various types of network participant?
 - Asset issuer
 - Network participant (signer)
 - Corda node administrator
 - Identity verifier
- Service provider, e.g., oracle
- Which participants require a node/identity?
- How many participants of each type will the network support?
- Define all the stakeholder roles from each organization that will be required for each participant to deploy a node and join a network, e.g, Procurement, IT Security, IT network, IT change management, IT App Dev team supporting the app portfolio related to the business network.

G.2. Define the Asset Model

- Asset(s) definition and important elements.
- Asset(s) lifecycle.
- Business process driving the asset(s) lifecycle among the participants in the target network.

G.3. Outline Legal and Regulatory Needs

- Identify legal and regulatory parties.
- Capture legal and regulatory requirements/boundaries.
- Assess and prepare for any intellectual property boundaries.

G.4. Establish the Network Financial Model

- What are the operating cost factors?
- Is network for profit? Cooperative with shared costs?
- How will the costs be shared? Or how will fees be determined?
- What is the initial funding required to start? (procurement involvement?)

G.5. Establish Data Privacy Policies

- What elements are sensitive?
- Who can create, modify, view which elements?
- Is data-at-rest encryption required?

G.6. Establish Data Retention policies

- Ensure clear guidance based on legal requirements for data retention needs.
- Is there a need to prune data that exceeds the data retention requirements?

G.7. Establish Data Resiliency and System Availability Policies

- What are the availability requirements?
- How much down time may be tolerated?

- Are their geo-political boundaries within which the data must remain?
- What is the recovery time SLA?

G.8. Assert Data and/or Processing Standards

We need to define how information, data, assets, etc., are represented within the business network. In some cases these will be proprietary representations, but in others we may wish to adopt national, international, or industry, standards such as ISDA or ACORD.

- How will data be represented, and how should it be interpreted?

G.9. Establish Transaction Validation Policies

- Policies driving CorDapp Contract code
- Policies related to asset model validations
- Policies specific to each party's independent validation

G.10. Establish Dispute Management Policies

- How are exceptions managed at:
 - Human interaction level?
 - Business processing level?
 - Smart Contract level? Is there a rollback where transaction is deemed invalid, or might there be some interim state recorded depending on the type of exception processing?

G.11. Establish On-boarding Policies

- What processes is needed to add a new party?
- Any regulatory requirements for adding a new party?
- What is the timing, the SLA requirement?

G.12. Establish Off-boarding Policies

- What processes is needed to remove a party?
- Any regulatory requirements for removing a party?
- What is the timing, the SLA requirement?

G.13. Establish Inter-Network Policies

There may be no need for inter-network interactions, but where they exist then we should give some thought as to any policies around their use.

G.14. Define Performance SLAs

- Acceptable ranges for the network operating hours:
 - Throughput (average vs peak)
 - Latency

G.15. Establish Change Management Policies

- Establish rules for how to agree on policy changes described in this section?

G.15.1. Business Network Role Changes

- Establish rules for managing BN role changes, the network governing body, e.g., is there an election process?

G.15.2. Business Network Legal Agreement Changes

- Establish rules for managing BN legal agreement updates.

G.15.3. Corda Platform Change Management Rules

- Establish policies for agreeing on platform updates.

G.15.4. CorDapp Change Management Rules

- Establish policies for agreeing on Cordapp code deployment, testing, versioning, and general lifecycle management.

O. Operations

O.1. Devise Node Configuration

- Node role: participant, network map, permission, notary, oracle, observer
- Node geographic location
- Node deployment host, i.e., on-premises, cloud (provider?), Corda Connect?
- Corda version
- Data store
- Does the data store provide the HA/DR capabilities to meet the BN resiliency requirements?
- Is there sufficient storage space to meet the data retention requirements?
- Is the node host/container appropriately configured to satisfy the performance SLA for the expected workload?
- Does participant already have a Corda node to which new Cordapps may be deployed?

O.2. Notary Configuration

- Location? Is location important?
- Protocol type, e.g., RAFT, BFT?
- Validating vs non-validating?
- Introducing or retiring notary – in Corda, by leveraging Notary-change transaction, it is possible to switch notaries, effectively introducing a new notary or deprecating, and ultimately, retiring their use, as needed.

O.3. Oracle Configuration

- Is the Oracle source verified?
- What are its configuration requirements?
- What are its data sources?

O.4. Create change management procedures

- Create and validate the change management procedures to meet the policies established by BNG. E.g. establish procedure for changing the consensus protocol

O.5. Establish Operational Times

- Satisfy System Availability policies.
 - What are the availability requirements?
 - How much down time may be tolerated?
 - What are the normal service hours of the network?
 - How will service interruptions (planned/unplanned) be communicated?
 - How will service interruptions be conducted?

O.6. Establish monitoring metrics

- What node metrics/alerts are required?
- What network metrics/alerts are required?
- What business metrics/alerts are required?

O.7. Define audit, report and documentation requirements

- What periodic network reports are required, and cadence?
- What periodic business reports are required, and cadence?
- What are the audit logging requirements?
- What documentation is needed?
- Who is responsible for creating/maintaining documentation?
- Where will documentation be available, accessible?

O.8. Define the network support parameters

- What are the first, second, third line support requirements, e.g., response times, escalation paths?
- Are there any support SLAs?

O.9. Define Data Retention and Pruning Procedures

- Based on the data retention guidelines, ensure the network has clear procedures to retain and prune data as needed, while being wary of data dependencies and maintaining the integrity of the network.

Architecture

Blah, blah.

Engineering

Note

Before reading any further, it is recommended to get familiar with the [Key Concepts](#) and [Corda Technical Whitepaper](#) first. Contents of this page assume high familiarity with Corda.

This article gathers together some guidelines, considerations and best practices for software engineers to keep in mind when designing a decentralised application on Corda. While recommendations are not mandatory, getting

familiar with them might reduce a cost of software upgrades or even prevent from having to redesign application from a scratch in the future.

Contents

Engineering	33
Foreword	34
Data model	35
Data privacy	35
Data updates	37
States	37
Contracts	37
On-chain vs off-chain	37
Contracts	38
Signers vs Participants	38
Determinism	39
Flows	39
Checkpointing	40
Corda Services	40
Flow upgrades	41
CorDapp structure	41
Contracts and States	41
InitiaitingFlows and InitiatedFlows	41
Build	42

Foreword

Even though Corda can be applied to a *huge* variety of usecases, like any other technology Corda doesn't solve *every problem in the world*. Developing for Corda requires some mind switching from classical centralised architectures. Developing for Corda - is designing a *network* rather than just an *application* as a solid piece of software.

Corda networks are constituted of a number of different participants, where each plays its own role(s) and might have *different* parts of the same application installed. For example a simple payments network on Corda might consists of a Money Issuer, FX Oracle and actual users. In essense they are 3 different actors, which all would have differnet CorDapps installed to their nodes.

Applications for Corda should be designed to be modular but still interoperable. Corda allows to split *private business logic* from an *API layer* into different CorDapps, and to install them separately onto the nodes of relevant participants.

Before jumping straight to the code, it is always useful to do some planning work ahead. Going *at least* through the following steps might save a lot of time and efforts in the future:

- Define actors. What type of participants are going to transact on the network? This will have an impact on a number of CorDapps you will need to produce.
- Define the asset model. Sketch a simple object or class diagram. Define relations between them.
- Understand privacy requirements. How should the data model be split between the actors? What visibility over the dataset should each of the actors have? Verify that the data model design fullfills the privacy requirements. As shown in one of the further sections, privacy requirements largely drive data model and transactions design. Keep in mind about *chain walkback* and that transactions are *irreversible*.

- Define transactions model. Is batching required to fulfill throughput requirements? Keep in mind that *business transaction* doesn't always have to be 1-to-1 to *Corda transaction*.
- Sketch sequence diagrams. Define interactions between participants. Verify that data and transaction model aligns with the flows. Sequence diagrams map really well onto Corda Flows.

The rest of this document is *not a tutorial*, but rather a set of *best practices* and *design patterns* for CorDapp developers to keep in mind.

Data model

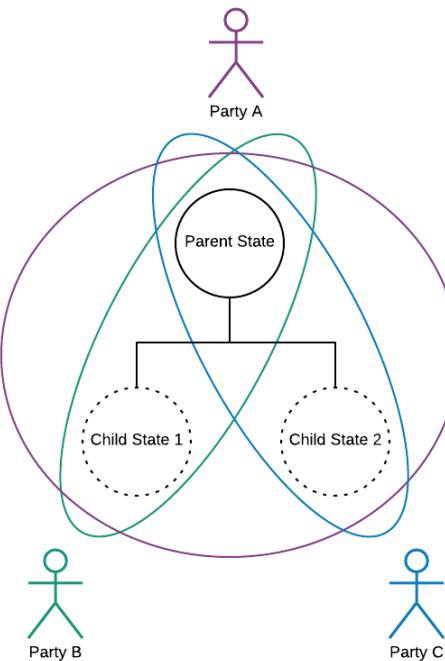
Noone would argue that getting the data model right is crucial for application's success. It becomes even more true on Corda, where data model and transactions design are largely driven by privacy, atomicity, verifiability and security requirements. In classical centralised architectures, dataset can be reshaped and filtered depending on the context, while in the Corda world, incorrectly designed contracts and transactions might lead to unwanted revelation of private data during transaction chain verification. However, by leveraging features and flexibilities offered by Corda's *data model* it is possible to fulfill even sophisticated requirements.

Data privacy

Data Privacy is one of the main concerns when designing DLT applications. Corda was built with privacy in mind with peer to peer networks, [flexible transaction model](#), [confidential identities](#), chain snipping, [transaction tear-offs](#), [non-validating Notaries](#) and future support of [Intel SGX](#). While these technics work very well out-of-the-box, sometimes data model has to be designed in a certain way to fulfill privacy requirements.

In a lot of scenarios, data model can be represented with a graph, where different actors have visibility over different branches. Well designed data model should offer benefits of both Corda (transactions, atomicity, privacy) and OOD (easy object graph traversal) in conjunction with data masking - i.e. giving different actors visibility over different branches, while keeping the whole object graph consistent.

Imagine the case, where ParentState, ChildState 1 and ChildState 2 need to be issued onto the ledger. All of the three states belong to the same object graph, however not each of them should be visible to all of the participants.



On the image above, *Party A* sees all three states, *Party B* - only ParentState and ChildState 1 and *Party C* - only ParentState and ChildState 2. Imagine the following class structure, where ChildState refers to its ParentState by linearId .

```

data class ParentState(
    val partyA : Party,
    val partyB : Party,
  
```

```

    val partyC : Party,
    override val linearId: UniqueIdentifier = UniqueIdentifier() : LinearState {
        override val participants = listOf(partyA, partyB, partyC)
    }

    data class ChildState(
        val partyA : Party,
        val partyB : Party,
        val parentStatId : UniqueIdentifier,
        override val linearId: UniqueIdentifier = UniqueIdentifier() : LinearState {
            override val participants = listOf(partyA, partyB)
        }
    }
}

```

To fulfill the privacy requirements, all three states should never appear on the same transaction, otherwise *Party B* and *Party C* will learn about the states owned by their counterparts.

Transaction 1 -> Party A issues ParentState with Party A, Party B and Party C **as** participants and signers
 Transaction 2 -> Party A issues ChildState 1 with Party A and Party B **as as** participants and signers
 Transaction 3 -> Party A issues ChildState 2 with Party A and Party C **as as** participants and signers

This approach will work well. Participants will be able to reconstruct relevant to them part of the object graph, by running multiple queries against their vaults. However, if the requirement says that *Party B* and *Party C* shouldn't know about neither identities nor even existence of each other, then described approach wouldn't work anymore, as *Party B*'ll become aware of *Party C*'s existence during the ParentState issuance, even if the both of them use *confidential identities*.

To address such privacy requirement, the following trick can be used. As you know, Corda commands can contain any arbitrary data. Regardless of the embedded data's nature, even if it's a valid Corda state, it's chain won't be *walked back* during the transaction chain verification.

```

data class MyCommand(
    val ledgerState : StateAndRef<ParentState> // <- this state's chain will not be "walked back"
) : CommandData

```

Now, lets make *Party A* to be the only participant of the ParentState and the only signer of its *issuance transaction*. Then, instead of including ParentState directly as an *input* into ChildState's *issuance transaction*, we can add *command* which would wrap ParentState's *issuance transaction*.

```

data class IssuanceCommand(
    val parentStatIssuanceTx : FilteredTransaction // <- this transaction will include Party A's signature as well as the actual Pa
) : CommandData

```

By doing this we will:

- Prevent ParentState from being consumed on ChildState issuance
- Give *Party B* and *Party C* verifiable prove that ParentState exists on the ledger
- Keep *Party B* and *Party C* unaware about existence of each other

Now, we have the following transaction structure:

Transaction 1 -> Party A issues ParrentState to himself, with himself **as** the only signer and participant
 Transaction 2 -> Party A issues ChildState 1 with Party A and Party B **as as** participants and signers, with IssuanceCommand
 Transaction 3 -> Party A issues ChildState 2 with Party A and Party C **as as** participants and signers, with IssuanceCommand

To prevent *Party A* from unilaterally reissuing ParentState without letting others know, we can utilise immutability pattern by including the following into the ParentState's contract.

```

class ParentStateContract : Contract {

    //...

    override fun verify(tx: LedgerTransaction) {
        if (!tx.inputs.isEmpty())
            throw IllegalArgumentException("Transaction can't consume any states!")
    }
}

```

```
//Other verifications
}
}
```

Data updates

Corda v3 is the **minimum** production-ready version of the platform. With a bespoke [serialisation framework](#), Corda v3 has introduced [wire stability](#) to allow different Corda node versions to coexist on the same network and to enable easier upgrades.

Corda has a separate well-defined procedures for upgrading States, Contracts and Flows, which all might and should have an independent upgrade cycles. Corda allows developers to split a single *logical* application into multiple CorDapps (jars), by which allowing each of them to be updated separately.

States

Past Corda v3, State classes can be upgraded without a need to update the ledger. Class evolution is supported by the serialisation framework and is documented [here](#).

Note

In the cases when the data model requires changes, which are not supported by the serialisation framework, a separate upgrade transaction might still be required.

Corda supports complex [vault queries](#). This is achieved by making a state to define a data schema, required by `QueryableState` interface. During initialisation, Corda automatically provisions a separate database table for each `QueryableState`'s schema (if it doesn't already exist). However schema changes are not handled by Corda automatically and are *have to be implemented by CorDapp developer*.

Note

Corda doesn't support database migrations out-of-the-box. When any of the state's schema changes, the CorDapp developer would need to provide a script to upgrade the database along with the updated CorDapp. For example, tools like [Liquibase](#) or [Flyway](#) might be used for database versioning.

It's always important to verify that evolution of State classes, defined by annotations, *matches evolution of database*, defined in scripts.

Contracts

In comparison to other blockchain platforms, Corda natively supports Contract upgrades. There are two ways of doing an upgrade: [implicit](#) and [explicit](#). With *implisit* upgrades, CorDapp developer can allow multiple implementations of the contract ahead of time by using constraints, which would allow the developer to benefit from painless upgrades in the future.

On-chain vs off-chain

It won't be a revelation to those who worked on legacy systems, that with time data model might get quite complex. A lot of new fields and relations are introduced through the years, which all contribute to model's complexity and sometimes make it barely manageable. Migrating big legacy data models onto the ledger might be a challenging task, which often leads to frustration in DLT as the technology.

However, not *the whole* datamodel has to be put onto the ledger in one go. Especially in the cases with legacy systems, application might benefit from bringing only a *part* of the model on-chain, while keeping the rest off-chain. To speed-up adoption, DLT can be introduced gradually, by replacing *some* of the existing layers step by step, but not the whole system in one go.

```

class SuperComplexDataModel( // <- Stored off-ledger
    val id : String,
    // other fields
)

data class MyState(      // <- Stored on-ledger
    val refDataId : String, // refers to the reference data by its id
    // other fields
) : ContractState

```

Inside **flows**, developers can combine off- and on-ledger parts by querying any external systems and using thirdparty libraries of their chose.

Note

One of the ways to assert correctness of the off-ledger data, would be to include it's hash into Corda transaction. When required, reference data's hash can be recalculated and compared with the one from the ledger.

Contracts

Corda Contracts are a crucial part of the *verification consensus* mechanism. Every Corda transaction contains at least one attachment with classes of the *referred* contracts and states. The idea behind this, is that any state's participant should be able to walk back transaction chain at any time, without having to have an access to the original CorDapp, where the transactions have been created from. It is very important to understand, that contracts - are a part of consensus mechanism, while flows - not, as flow-logic can't be replayed in the future.

Some developers are tempted to put transaction verification logic inside the flows, as they are less restrictive than contracts. This would be not a good practice for the reasons described above. As a rule of thumb, all logic related to verifying transaction as an independent unit, shuld go into *contracts*.

Signers vs Participants

Corda separates owners of a state from the *signers* of a transaction, whcih involves this state. This allows multiple parties to come to the consensus about a complex atomic multi-state transaction, while in the same time to maintain a fine-grained ownership of each of the states inside this transaction. State owners are specified as participants on the State level, while signers are specified for each Command on *transaction* level. It's always a responsibility of the CorDapp developer to match signers vs participants inside Corda Contract. Not having done that would lead to a flaw, where participants would be able to consume *not theirs states*.

```

data class MyState(val participantA : Party, val participantB : Party) : ContractState {
    override val participants = listOf(participantA, participantB)
}

class MyCommand : CommandData

class MyContract : Contract {
    override fun verify(tx: LedgerTransaction) {
        val myState = tx.outputs.single { it.data is MyState }.data
        val myCommand = tx.commands.requireSingleCommand<MyCommand>()
        requireThat {
            // veririfies that all signers are participants
            "Participant have to match signers" using (myState.participants.map { it.owningKey }.toSet() == myCommand.signers)
        }
        //... OR
        // veririfies that signers contain all participants
        "Signers have to contain all participants" using (myCommand.signers.toSet().containsAll(myState.participants.map { it.owningKey }.toSet()))
    }
}

```

```
// other verifications
}
}
}
```

Determinism

Contracts are executed in the deterministic jvm (DJVM), which imposes some constraints on contents of the verify() function. It should contain no calls to non-deterministic APIs, such as Date, Random, IO and others. Any calls to non-deterministic APIs would fail the contract execution. Code inside the verify() function can only access internals of the LedgerTransaction as well as *deterministic* APIs of JDK. No thirdparty libraries are visible to the Contracts, unless they are physically bundled into the jar with *contracts-and-states*.

Note

It's important to emphasise the fact, that a thirdparty library, included into contracts-and-states jar, becomes a part of the verification consensus. It is highly recommended for CorDapp developer to get absolutely familiar with the library's internals and to make sure that he trusts each and every aspect of it. Any thirdparty code to be used within the contracts, should be verified for determinism as well.

[I DON'T KNOW IF WE SHOULD MENTION ABOUT JSR303 APPROACH WE TOOK WITH B3I]

Flows

Corda Flows Framework is an abstraction layer over the underlying messaging system, which also provides *checkpointing*, *crash recovery*, *session management* out-of-the-box. Flows are *not* a part of the *consensus mechanism* in Corda. Any verifications, performed inside the flows, will be run *only* during transaction negotiation and *won't* be replayable at the later date during transaction chain walk-back.

```
data class NumberState(val number : Int) : ContractState

@InitiatedBy(MyInitiatingFlow::class)
class MyInitiatedFlow(val session : FlowSession) : FlowLogic<Unit>() {

    @Suspendable
    override fun call() {
        // ...
        val numberState = session.receive<NumberState>().unwrap {
            if (it.number < 10) { // <-- this constraint will be checked only during transaction negotiation, meaning that it won't be re
                throw IllegalArgumentException("Number should be less than 10")
            }
        }
        val builder = TransactionBuilder(notary)
            .addOutputState(numberState, NUMBER_CONTRACT_NAME)
        //...
    }
}
```

Note

While Flows provide a great mechanism to build multiparty communications, all consensus-related logic should be placed into the Contracts.

Note

While Corda communications are always peer-to-peer, it is possible to have multiple sessions with different parties opened inside the same Flow.

Checkpointing

Under the hood, Corda uses [Qasar](#) library to transform user-written flows into asynchronous persistent state machines. Every time, when network communication occurs, Flow persists its state into the underlying database and releases the resources it holds. During checkpointing, all objects referenced from the stack are serialised into a blob and persisted into the database as a part of the Flow's state.

```
@InitiatingFlow
class MyInitiatingFlow : FlowLogic<Unit>() {

    @Suspendable
    override fun call() {
        // ....
        val largeObject = readLargeObjectFromDisk() // <-- reads very large object from a disk and keeps a reference to it on the stack
        val session = initiateFlow(counterparty)
        session.sendAndReceive<String>(movie) // <-- flow will be checkpointed. Large object will be serialised and persisted in the database
        // ...
    }
}
```

Note

To avoid potential performance issues, it's a good practice to not to keep references to large objects visible to checkpointing.

Corda Services

@CordaServices can be used to implement a singleton pattern, when an instance of a shared service is used by multiple flows.

```
class MyService : SingletonSerializeAsToken() { <-- Singleton services should extend from SingletonSerializeAsToken
    fun doStuff() {
        // ....
    }
    // ...
}

@InitiatingFlow
class MyInitiatingFlow : FlowLogic<Unit>() {

    @Suspendable
    override fun call() {
        // ....
        val service = serviceHub.cordaService(MyService::class.java) // <-- the same logical instance of service is returned even if multiple flows
        // ...
    }
}
```

This pattern can be used for example to load CorDapp's configuration from a disk.

```
class ConfigurationService : SingletonSerializeAsToken() {
    fun getApplicationConfiguration(): Map<String, Any> {
```

```

    // read configuration from a disk and cache in a local variable
}
}

@InitiatingFlow
class MyInitiatingFlow : FlowLogic<Unit>() {

    @Suspendable
    override fun call() {
        // ....
        val configurationService = serviceHub.cordaService(ConfigurationService::class.java) // <-- Configuration will be red from
        // ....
    }
}

```

Flow upgrades

As mentioned earlier, flows have a separate upgrade cycle. More information about flow versioning can be found [here](#)

CorDapp structure

This section contains design patterns and best practices for organising CorDapps.

Contracts and States

Contracts and States should be kept in a separate from Flows jar file, out of the following reasons:

- *Performance issues.* Contracts and States classes are attached to transactions in Corda. Even though attachments are cached locally by nodes, a potentially large jar file would have to be transferred *at least* once to each of the participant. Large *contracts-and-states.jar* might also cause performance issues in Flows during checkpointing and LedgerTransaction resolution.
- *Privacy issues.* *contracts-and-states.jar* might potentially be shared with all other participants on the network and hence shouldn't contain any private or sensitive data.
- *Upgrade issues.* Contracts, States and Flows all have separate upgrade cycles, as described in the previous sections. Bundling all of them together will largely reduce upgrade flexibility.

InitiatingFlows and InitiatedFlows

Some CorDapps are required to provide APIs to other applications on Corda network. In that case, CorDapp with Flows needs to be split into 2 jars - one with `@InitiatingFlows` and the other with `@InitiatedFlows`. CorDapp containing `@InitiatingFlows` would be shared with all thirdparties and would effectively become an *API*. CorDapp with `@InitiatedFlows` would be deployed to the owner's node only. By doing so, the CorDapp owner would keep his business logic private to himself while would also provide others with a well defined API to his application.

Imagine an FX Oracle. The following classes would go to the API CorDapp.

```

@CordaSerializable
data class FXQuote(val from : Currency, val to : Currency, val quote : Double)

@CordaSerializable
data class FXQuoteRequest(val from : Currency, val to : Currency)

@InitiatingFlow
class FXOracleApiFlow(val request : FXQuoteRequest) : FlowLogic<Unit>() {
    companion object {
        val ORACLE_NAME = CordaX500Name.parse("O=My FX Oracle,L=London,C=UK")
    }
}

@suspendable

```

Operations

```
override fun call() {
    val oracleParty = serviceHub.identityService.wellKnownPartyFromX500Name(ORACLE_NAME)!!
    val oracleSession = initiateFlow(oracleParty)
    // API package doesn't know anything about where the quote is coming from and what the responding flows are.
    val fxQuote = oracleSession.sendAndReceive<FXQuote>(request).unwrap {
        // ....
    }
}
```

While these classes would go to the CorDapp, private to the Oracle.

```
@InitiatedBy(FXOracleApiFlow::class)
class FXOracleInternalFlow(val session : FlowSession) : FlowLogic<Unit>() {

    @Suspendable
    override fun call() {
        val request = session.receive<FXQuoteRequest>().unwrap { it }
        // .. get fx quote from internal systems
        session.send(FXQuote(request.from, request.to, quote))
    }
}
```

Oracle's node would need to have both of the CorDapps deployed for it to know how to reply to `@InitiatingFlows`, while other participants would be aware of the API CorDapp only, which doesn't contain any sensitive logic.

Build

Detailed build recommendations are available [here](#). If your CorDapp has a dependency conflict with Corda, would worth to have a look into the [Maven Shade Plugin](#).

Operations

Blah, blah.

Frequently asked questions

What is this site for?

This site sets out to explain how to use Corda to solve real problems. It doesn't tell you what Corda is, nor does it seek to describe how it's built, but instead focuses on how to use it to the best of its capabilities.