Technical guidelines for implementation

CHAPTER 03

A
Getting the infrastructure right: 10 key technical steps

Blockchain can be implemented in many ways either for a single use-case or for multiple use-cases. While the implementation could take many approaches in respect of technical considerations and trade-offs, the standard procedures, processes and methods of design, development and deployment do not differ significantly in the context of best practice. The key technical steps of implementation include identifying the use-case, testing, integrating portals and user interfaces, integrating the blockchain with existing infrastructure, and security protocols. Figure 6 shows the ten key technical steps in more detail.

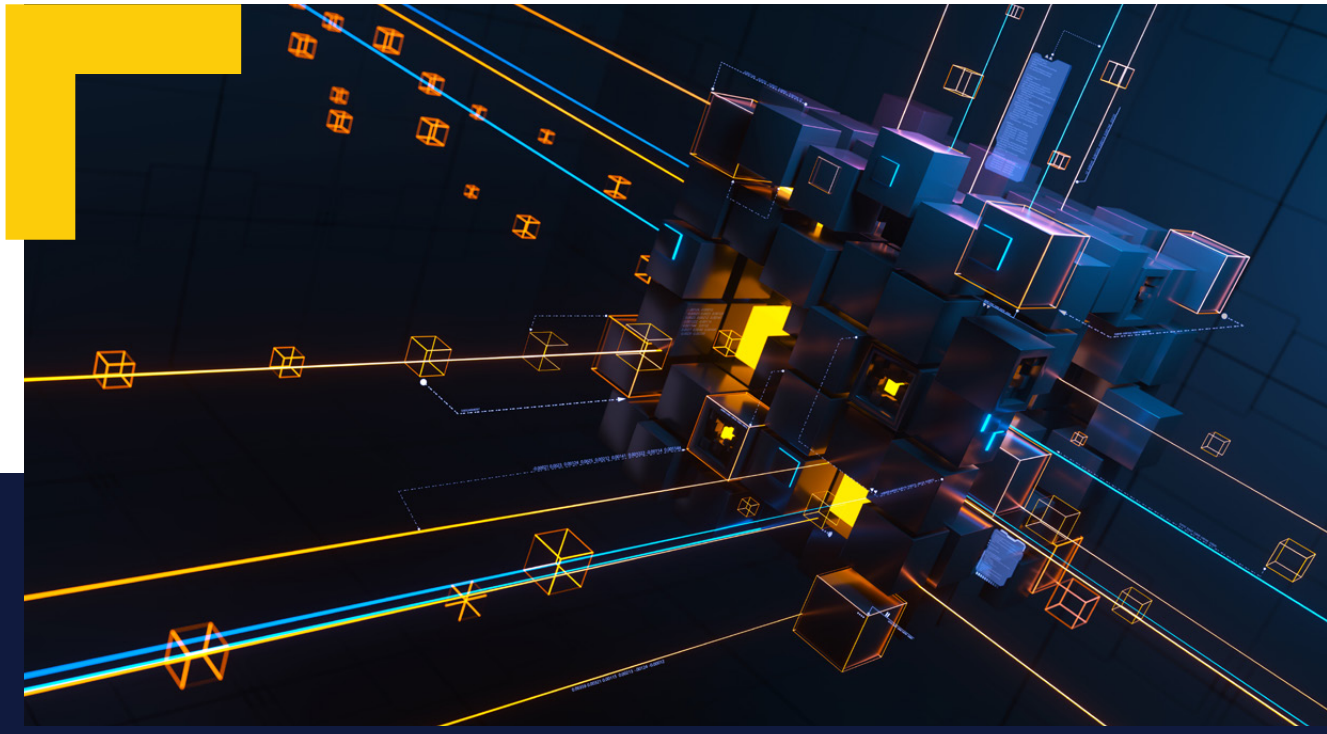
Figure 6. The 10 key technical steps for the blockchain implementation process

Source: Compiled by ESCWA.

1. Identify the needs, define the requirements and scope the use-case

To implement a blockchain for any purpose, defining the use-case based on the identified trade facilitation needs is the first and perhaps the most important step. Getting things right at this step will influence other subsequent steps such as the architectural design, and choices around speed and security as well as permissions and privileges of members. Identifying the specific problems that the blockchain is intended to solve and scoping the technical design accordingly will help the implementing body avoid difficulties later on in the implementation process.

A blockchain that is intended to process payments of duties, for example, may need high speed and high sovereignty at the core architecture in order to avoid delays and infiltration, while a blockchain for trade audits would require higher storage and data protection capabilities. Identifying the use-case for the blockchain and scoping the problem it aims to solve informs the architectural design, as well as, various aspects of the policy considerations. Identifying and defining the purpose of the technology within the trade environment is a foundational requirement for the subsequent technical steps in building the blockchain. In the end, this step will largely inform the degree of decentralization, security and speed of the network as well as the adopted architectural parameter and technical specifications of the protocol.



While a blockchain could be implemented to address multiple trade facilitation needs around trade audits, risk management, quality assurance, transparency and data validation, finding a balance between performance, security and sovereignty without major compromises is crucial.

2. Select the blockchain protocol and platform

The proliferation of different blockchain protocols over the years has brought to light the technical trade-offs that have to be made in deciding on the choice of protocol for a blockchain use-case. The chosen blockchain protocol necessarily influences the capabilities of the technology implemented, given the trade-offs to be made between speed, sovereignty and security. A well-distributed and largely decentralized blockchain protocol tends to be slow and lack speed but has high security. On the contrary, private/consortium blockchain networks tend to be fast in speed, high in storage, but lower in security and resilience. Furthermore, proof of work (POW) protocols such as that of Bitcoin, tend to be slower than proof of stake (POS) protocols such as those of Ethereum or Cosmos.[1] Thus, depending on the use-case envisaged and defined in step one, some protocols and architecture may be unfit for the use-cases in trade measures and others will be more suitable. Key technical considerations largely cover specifications around speed, quality assurance, data integrity, storage, resilience and security. Collaboration with the private sector, industry and academia will be useful in building the architectural details of the implementation process.

3. Design and implement the architecture

Designing the architecture of the blockchain is the first hands-on technical step of putting the network together that serves as the foundation of the technology. Once the use-case is known and the protocol is chosen, the technical design process entails a deeper and more detailed step of putting together the core of the blockchain network. The technical design step involves the network architecture, consensus mechanism, node configuration, levels of authority of network participants, emergency restoration and recovery processes, cryptographic functions, risk management procedures, digital signature protocols and other parameters. It is recommended that at this stage, private sector consultants and experts from industry collaborate with the Government in the execution process. Blockchain is relatively complex and requires skills in multiple fields to be implemented. Most Governments, especially those of developing countries, still lack the expertise and skills within their agencies to execute all the technical details. Given the lack of in-house expertise to execute the deeply technical aspects of blockchain technology, collaboration with the private sector and industry experts is advised for a successful implementation process.

4. Testing

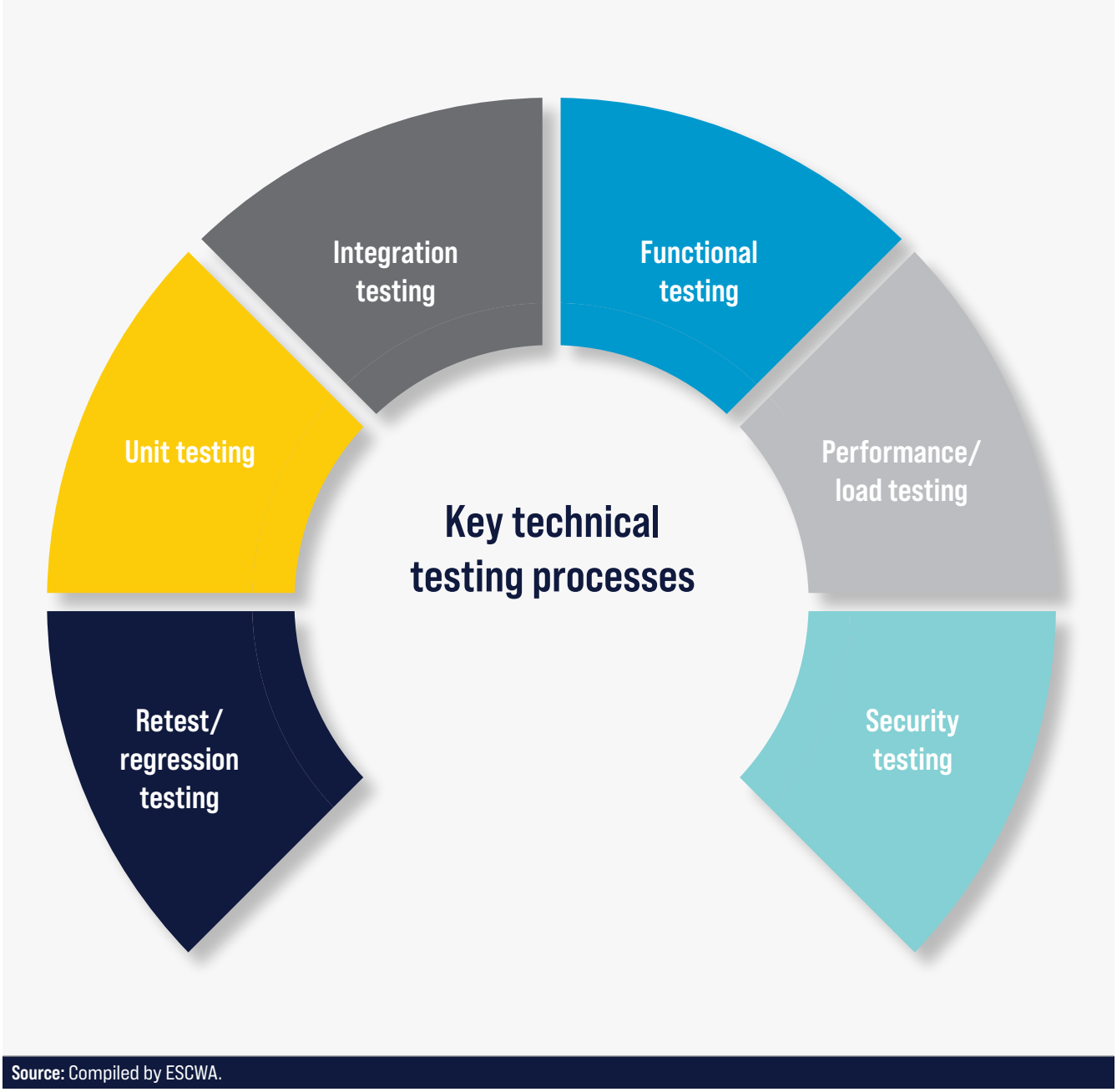
The complexity of blockchain brings with it significant chances of bugs, errors and code malfunctions. Hence, testing is a key step of the design, development and deployment process. This step usually entails unit and process assessments that allow for components as well as constituents of the code to be tested for risks, vulnerabilities and bugs. Testing is done within simulated environments in the form of testnets. It examines the network's functionality, security and stability, simulates real-world scenarios and ascertains that the blockchain will work as intended and without major flaws. During this phase, all major network issues are identified and fixed. Before implementing the technology for trade processes, customs, authorized economic operators (AEOs), traders and other agencies should be involved in the

1. More details about these protocols and how they differ from each other can be found in the Global Report on Blockchain and Implications for Trade Facilitation Performance, published by the United Nations Conference on Trade and Development in 2023.

key testing steps to ensure that the blockchain addresses all key user requirements and meets international standards. Testing allows the implementing body to identify major issues especially around security, compliance and interoperability and fix them ahead of deployment. This is an essential step in the development process of any blockchain and would usually entail key sub-processes as shown in figure 7 and detailed in box 1.

The complexity of blockchain brings with it significant chances of bugs, errors and code malfunctions.

Figure 7. Key components of testing in the blockchain deployment process



Source: Compiled by ESCWA.

Box 1. Seven key components of testing in the deployment process

Unit testing: involves testing individual components and modules of the blockchain network to ensure that they work as intended. At this stage of testing, automated testing frameworks and tools can be used to identify and fix issues in the early parts of the development process.

Integration testing: involves testing how different components of the blockchain interact with each other to ensure they do so as intended. This step ensures that the constituents of the program work as a whole and that there are no compatibility issues between different components.

Functional testing: involves testing features and key functionalities to ensure that they work in accordance with the original design and intent and that the key technical requirements specified in the design phase are achieved. This step tests the entire process and confirms that the blockchain performs as expected and can handle the technical workload that it is expected to handle.

Performance/load testing: examines whether the blockchain can handle the processing volume, workload, speed and multi-user interactions. This step investigates the computational strength of the blockchain and its ability to handle the expected user traffic in order to fix any performance issues that could affect user experience beforehand.

Security testing: checks for security vulnerabilities and bugs that could affect security functionalities and any other potential risks of unauthorized access, network disruption and insecurities. This step aims for the protection of the network's sensitive data.

User acceptance testing: involves testing the blockchain with real users to ensure that the design meets stakeholder expectations, delivers a good user experience and meets user capabilities. This step identifies usability bottlenecks that were not captured in the earlier testing phases but may be critical. The user feedback and stakeholder inputs help with designing a user-friendly blockchain protocol.

Retest/regression testing: involves retesting the blockchain's entire process after making changes and updates to fix vulnerabilities identified in the preceding testing phases. This step is taken to ensure that the updates and changes do not introduce new issues and vulnerabilities into the existing functionality.

5. Develop smart contracts

Smart contracts are self-executing codes, business logics and programs that are stored on the blockchain for specific logic and use-case purposes. While the blockchain itself is simply a distributed database with redundancies, smart contracts constitute the executive functionalities of the blockchain. They can serve as middleware between the blockchain database and the user-facing portal of the Government's trade facilitation portal, or they can simply be used to process logic between one or more functionalities. They could further serve as the process logic that provides access credentials, authorization, rights and privileges to users within a blockchain ecosystem. Smart contract design is a particularly important component of the blockchain design process. While a blockchain database could be a general purpose infrastructure for many use-cases, smart contracts allow for specific use-case designs and serve as compartmenting infrastructure to deploy individual

tasks and specific functionalities on the blockchain. For example, on a single blockchain, different smart contracts could be deployed to handle processes such as managing the digital identities of customs officers and AEOs, tracking and verifying trade document flows for audits, and automating authentication for the quality assurance of trade certificates and permits, all with the aim of meeting the Government's trade facilitation needs. As process-logic software used to define the rules of the network and automate processes, smart contracts are thus particularly useful in attributing and compartmentalizing user rights and privileges, levels of authority and user access controls, as well as conditions for the revocation of such user rights and privileges and undertaking risk mitigation mechanisms within the network.

6. Deploy the blockchain

Deploying the blockchain within a production environment is a stage when the network goes live, begins to process records, produces data pockets, creates data chunks (called blocks) and a digital history of the blockchain network. It usually involves a coordinated process of synchronization of computational units (servers/nodes) which are part of the network until a simple majority have been established on what can be considered as common agreement of the true state of the network records. Once the genesis block is up, running and producing further blocks, the network can be considered stable and safe and can then deliver the data reliability required in terms of meeting integrity requirements. This stage of the implementation processes comes with design and policy choices. For example, the implementing body could deploy the network on sovereignly owned physical computational infrastructure such as servers (usually called a bare metals approach) or opt for a cloud-based service provider like Microsoft Azure or Amazon Web Services. Choosing a cloud-based service means the implementing body doesn't own the infrastructure on which the network runs and thus gives up a level of sovereignty, although this choice may be cheaper. The implementing body may choose a third option where the blockchain is deployed using a Blockchain-as-a-Service (BaaS) platform such as the European Blockchain Services Infrastructure platform or the Blockchain-based Service Network in China, which allow any entity to deploy their own sovereign blockchain application without owning the digital infrastructure on which the blockchain runs. The implementing body gives up a level of autonomy and sovereignty with the use of these cloud-based option as well.

7. Design security protocols

The most significant value proposition of blockchain is data integrity. Thus, when a blockchain experiences unauthorized intrusion and data tampering, it causes numerous problems regarding re-establishing the true state of records on the blockchain and usually requires a network reorganization which can be resource intensive. Security breaches of

The implementing body gives up a level of autonomy and sovereignty with the use of cloud-based options.

blockchain that require network reorganization in order to clean the erroneous data from the network also mostly require special expertise, something most Governments may not readily have at hand. Implementing security protocols is thus critical for the overall network health, data integrity, as well as stability and reliability of the information that is stored on the blockchain. These security protocols include technical measures to prevent attacks and unauthorized access through data encryption, reliable private key protection mechanisms, access controls, risk prevention and mitigation protocols such as activity logs, procedures on emergency updates, code reviews, security patches, network halts and reorganization and restart procedures. These ensure network protection to prevent unauthorized tampering, access and attacks on data stored on the blockchain network.

8. Design and integrate portals and user interfaces

Although not technically part of the blockchain itself, user-facing portals and interfaces allow the average user to interact with the blockchain in a less technical manner. They also allow the business logic and process flow of the use-case applications of the blockchain to be experienced by the average user without deep technical expertise. Thus, they form the most critical part of the user experience and greatly influence the perceived user-friendliness, utility and value of the blockchain. As a complementary component of the business side of the blockchain, they are normally deployed as separate code with the sole function of ensuring that the user experience of the blockchain is optimal and efficient. This separate code can usually be designed by a separate team whose expertise lies in user interfaces and user experience but who, nevertheless, have good understanding of the technical primitives of blockchain and Web 3.0 in general. Implementation of these components can usually be done in a conventional Web 2.0 fashion.

9. Integrate with existing infrastructure

Implementing blockchain should not be done in isolation of the existing trade facilitation digital infrastructure. Doing this will create data silos and lead to suboptimal outcomes. Integration with existing infrastructure should be at the centre of the implementation process from the start. Given that most Governments already have existing trade facilitation solutions such as trade portals, trade reform trackers and single windows, a blockchain should only help improve specific trade facilitation measures that are not already handled well by other conventional legacy systems. Thus, the blockchain implementation process must be done in a manner that ensures integration with existing legacy infrastructure and allow for the possibility of future integrations. For successful integration with existing infrastructure, the implementing body should be guided by the steps set out in box 2 and shown in figure 8. There are also three key forms of integration, as shown in figure 9 and detailed in box 3, and these forms can be combined.

Given that most Governments already have existing trade facilitation solutions such as trade portals, trade reform trackers and single windows, a blockchain should only help improve specific trade facilitation measures that are not already handled well by other conventional legacy systems.

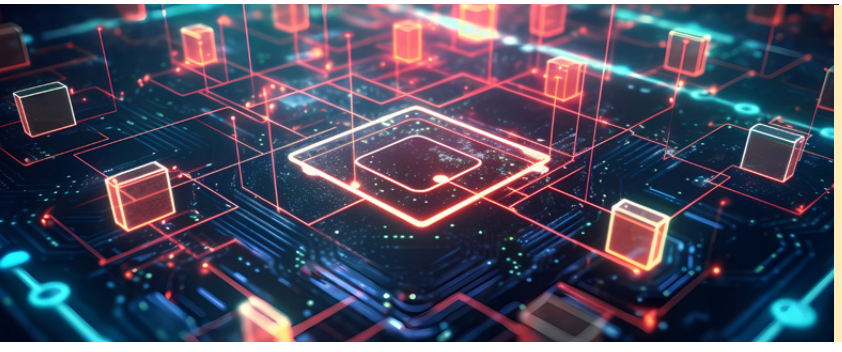
Box 2. Four key steps for successfully integrating blockchain with existing trade facilitation infrastructure

Define the purpose of integration: Defining the purpose of the integration will inform technical design choices around features and ensure proper alignment of the processes with the Government's overall trade facilitation goals. Depending on the targeted trade facilitation needs, the blockchain may be integrated in three possible ways: as a solely blockchain-backed utility, as a purely legacy digital system, or as a combination of legacy and blockchain back-ends where some application logics use a blockchain data back-end, while other logics lead to a conventional database back-end.

Design the technical requirements of integration: This includes the technical designs that will guide the design and implementation process of the integration. The design of technical specifications, connectivity systems and transport components that convey data to and from the blockchain environment and legacy infrastructure, as well as the overall architecture and user-facing portals, including the application programming interfaces (APIs) endpoints, data models and integration points, are put together at this stage. This step would usually involve multiple teams of the blockchain implementing body and the legacy technology groups to ensure that the integration is both technically feasible, functionally desirable and meets performance requirements.

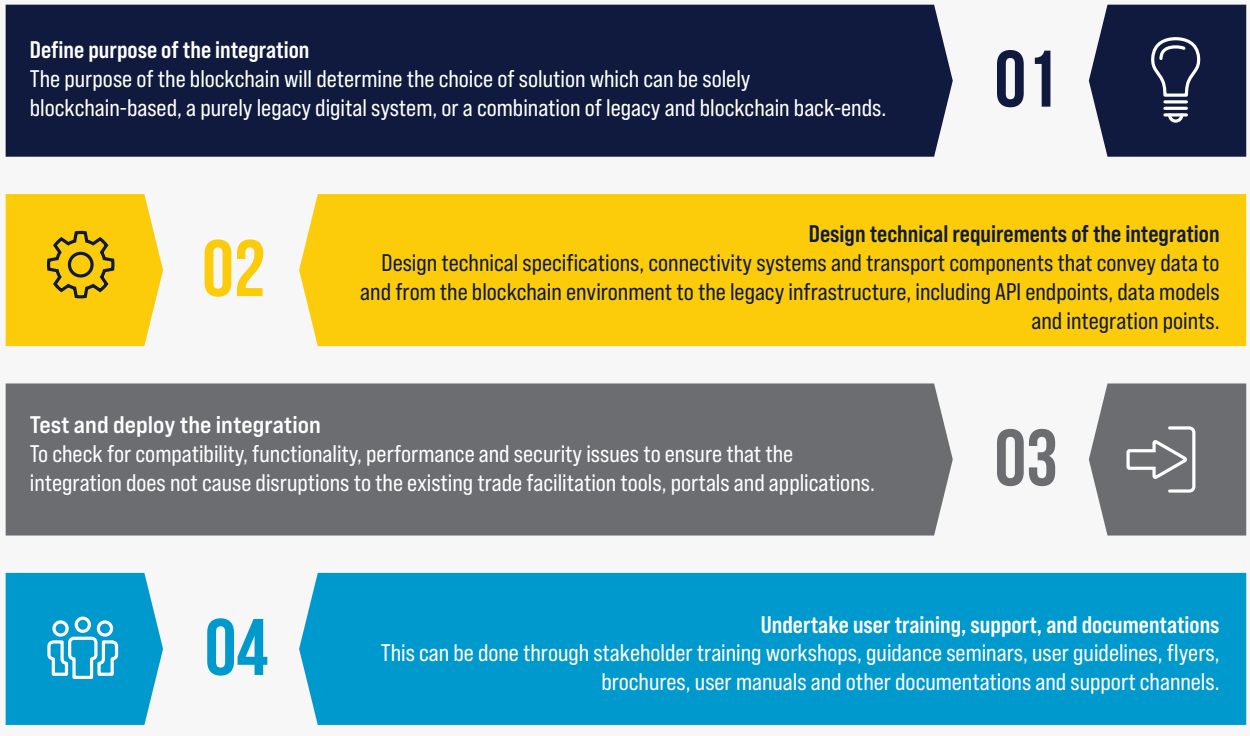
Test and deploy the integrated blockchain: As in all software development processes, the integration process of different infrastructures must be thoroughly tested and implemented after the architectural designs, requirements and feasibility are assessed. This generally involves testing code in a sandbox environment by the key teams in charge of the implementation for compatibility, functionality, performance and security to ensure that the integration will not cause any disruption to the existing trade facilitation tools, portals and applications. Once testing has gone smoothly, the integration can be deployed in a production environment and monitored closely for issues by tracking performance metrics, user feedback, event logs, security incidents and activity logs.

Provide user training, support and guidance documents: Once integration is complete, depending on the level of changes to the user-facing portals, interfaces and applications, user training and support for the new features will ensure smooth adoption and user acceptance. This step also becomes very important if new functionalities have been added to the existing application logic. Training and support can be provided in the form of stakeholder training workshops, guidance seminars, user guidelines, flyers, brochures, user manuals, other guidance documents and communication channels. Beyond continuous user support, the overall performance, functionality and security of the integrated infrastructure must be continuously monitored with regular reviews, upgrades and updates based on user feedback, evolving process requirements and emerging trade facilitation needs of the Government.



Implementing blockchain should not be done in isolation of the existing trade facilitation digital infrastructure.

Figure 8. The four key steps to get integration right



Source: Compiled by ESCWA.

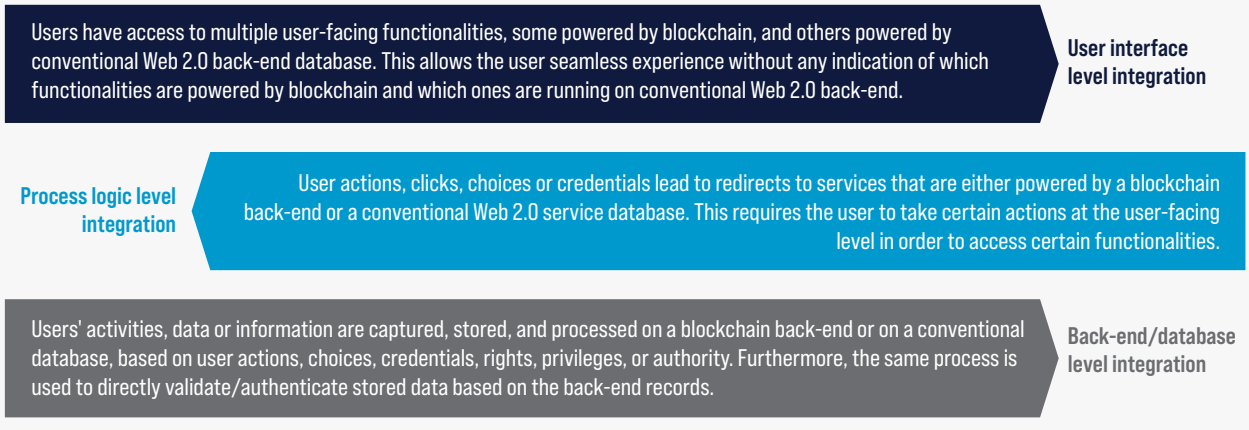
Box 3. Three key forms of integration

On a practical level, the way legacy systems and the blockchain ecosystem are integrated is worth addressing. Integration can take the form of various vertical and horizontal flows between the two ecosystems. For example, the blockchain back-end could also partly serve as the back-end of a conventional Web 2.0 application (a vertical flow), or both legacy systems and blockchain solutions could work in parallel for specific trade facilitation measures (a horizontal flow). The three key forms of integration are:

- Integration at the user-facing level: This is where users have access to multiple user-facing functionalities, some powered by blockchain, and others powered by a conventional Web 2.0 back-end. This provides the user with a seamless experience where they will not know which functionalities are supported by blockchain and which are powered by a Web 2.0 back-end. For example, a customs officer may use the same portal to approve the license of an AEO, a process recorded on a blockchain, as well as approve a trade declaration, which may be recorded on a conventional Web 2.0 database.
- At the process logic level: This is where, depending on the user actions, choices or credentials at the front-end, the user is directed to a particular service powered by a blockchain back-end or to a conventional Web 2.0 service. This option requires the user to take certain actions, make choices or enter certain credentials to be redirected to an application logic supported by a specific back-end.

- Integration at the back-end and database level: This is where, based on user actions, choices, credentials or privileges, the user's activities are captured, stored and processed on a blockchain back-end, or all processes happen on conventional database. For example, a trader may be allowed to use a blockchain-based Government service to independently authenticate the validity of a trade certificate issued by a Government agency in order to prevent fraudulent activities or to detect counterfeits. In this case by entering a certificate serial number or scanning a barcode/Quick Response (QR) code, the user gets validation from a blockchain-based records system or a legacy database. Furthermore, an AEO may be allowed to sign and store digital signatures on a blockchain network that allows for any future post-clearance audits to be able to trace activities to the AEO, and this will make quality controls easier for the Government and reduce trade risks. This could take place within a portal that supports both blockchain-based data and conventional database systems, with multiple process logics designed into it. Depending on the chosen purpose, the user would be directed accordingly to a blockchain back-end or to a conventional Web 2.0 back-end for validation and authentication of the information.

Figure 9. The key forms of integration for blockchain solutions and legacy systems



Source: Compiled by ESCWA.

10. Monitor, evaluate and maintain the blockchain and digital infrastructure

Once integration has been well-executed, regular monitoring and maintenance is required of the digital solutions of the blockchain, the complementary applications and the pre-existing digital infrastructure both in house and at the level of the network. The implementing body must cultivate a culture of regular evaluation and updates to resolve any functional, security and performance issues that emerge from time to time. Network health is critical for the continuous functionality and usage of the blockchain. Consequently, the implementing body must constantly assess activity logs to ensure that there are no network attacks or unauthorized access. The implementing body must also ensure that access controls are constantly improved to keep the network well-protected and its data integrity guaranteed. Monitoring the network for any issues and performing regular maintenance tasks, such as network upgrades and software updates can be done both as preventative measures but also as mitigative measures. Also, as the Government's trade facilitation needs evolve, new technical functionalities may be needed to meet the new needs. Stakeholders and users must also be constantly informed about digital hygiene and individual practices that keep data safe and protected. In table 4 the key technical implementation steps are summarised.

Table 4. Summary of key technical steps for implementing blockchain for trade facilitation

Key technical Step	Key deliverables
Identify needs, define requirements and scope the use-case	■ Needs assessment ■ Readiness assessment ■ Use-case scoping ■ Stakeholder coordination
Select blockchain protocol and platform	■ Defined use-case suitability, feasibility and performance requirements ■ Comparison of required specifications with existing design options ■ Choice of protocol design/architecture that meets design requirements or combination of various protocol features that meet the use-cases envisaged
Design and implement the architecture	■ Network architecture ■ Consensus mechanism ■ Node configuration ■ Participant authorizations ■ Emergency restoration and recovery processes ■ Risk management procedures ■ Digital signature handling procedures
Testing	■ Unit testing ■ Integration testing ■ Functional testing ■ Performance/load testing ■ Security testing ■ User acceptance testing ■ Retest/regression testing
Develop smart contracts	■ Middleware smart contracts ■ Process logic smart contracts ■ Access control smart contracts ■ User credentials, authorization, rights and privilege smart contracts
Deploy the blockchain	■ Decision on infrastructure set-up (cloud-based, platform-based or bare metals approach) ■ Launch main net ■ Get genesis consensus ■ Fully synchronize nodes and record production system
Design security protocols	■ Data encryption ■ Private key protection ■ Access controls ■ Application activity logs ■ Emergency update procedures ■ Procedures for network halts, reorganization and restarts
Design and integrate portals and user interfaces	■ User-facing portals, applications and solutions ■ Supportive technical procedures for digital signature processes and key handling
Integrate with existing infrastructure	■ Feasibility, functionality and performance assessments of the integration process ■ Technical specifications around connectivity, transport and architecture including APIs, endpoints, data models and integration points ■ Integration of legacy systems (trade portals, trade reform trackers and single windows)
Monitor, evaluate and maintain the blockchain and digital infrastructure	■ Regular maintenance ■ Network upgrades ■ Security updates ■ User digital hygiene ■ Data safety practices ■ New feature upgrades

Source: Compiled by ESCWA.

B
A work breakdown structure for the implementation process

It is important to break down the work into deliverables and tasks to be caried out by the implementing body in order to achieve the ten milestones elaborated above. This is considered the work breakdown structure (WBS) which is a technical decomposition of the implementation process into smaller, more manageable tasks.

It helps the implementing body properly scope the process, identify key requirements and organize resources efficiently. The technical breakdown of the design, development and adoption process of blockchain and its accompanying solutions are demonstrated as six broad steps in table 5 below.

Table 5. Technical work breakdown structure for blockchain implementation

Step	Implementation components
Commencement	■ Define the goals and objectives of the use-case. ■ Identify and define key technical and policy stakeholders. ■ Develop project implementation guidelines and other key documents. ■ Identify implementing partners.
Planning	■ Define the project scope of the use-case(s). ■ Identify functional, security and performance requirements. ■ Develop an implementation plan. ■ Design the implementation schedule with key timelines. ■ Create an implementation budget and resource limits. ■ Define project teams, roles and working relationships. ■ Pre-screen key implementing partners and service providers.
Design	■ Conduct user requirement analysis. ■ Define systems performance requirements. ■ Develop functional specifications. ■ Select the blockchain protocol. ■ Develop the network architecture. ■ Create design specifications for both the front-end and back-end. ■ Design the user interface and portal workflows. ■ Create a security protocol. ■ Develop code, including libraries, frameworks and modules. ■ Develop blockchain application-level modules.
Testing	■ Conduct testing, including unit testing, integration testing and systems testing, as well as security, functional, performance, user acceptance and regression testing
Deployment	■ Develop a full deployment plan that defines the order of execution. ■ Create a protocol for the network launch process. ■ Develop system requirements for foundational infrastructure. ■ Conduct stakeholder and user training. ■ Develop contingency plan for bottlenecks, halts and restarts. ■ Launch network and be prepared for unforeseen challenges.
Maintenance	■ Design a software maintenance plan. ■ Develop a procedure for updates, upgrades, halts and restarts. ■ Create a protocol for activity logs. ■ Carry out regular bug fixes. ■ Conduct regular updates and upgrades. ■ Undertake user support and training.

Source: Compiled by ESCWA.