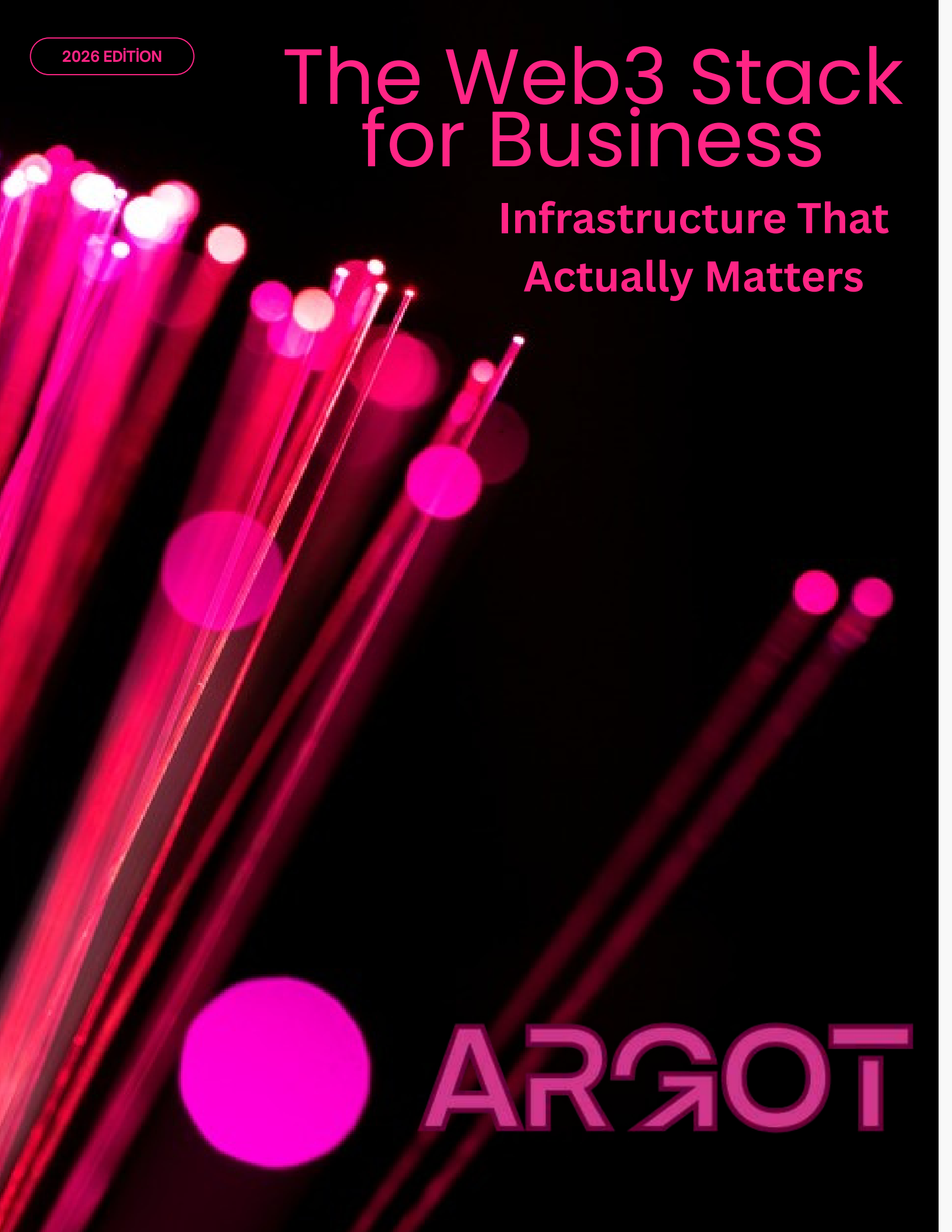
2026 EDITION

The Web3 Stack for Business

Infrastructure That Actually Matters



ARGOT

Executive Summary

Web3 is often discussed as a collection of individual technologies: blockchains, wallets, smart contracts, tokens, stablecoins, digital identity, and decentralized applications.
For businesses, that is the wrong way to think about it.

These technologies increasingly operate as parts of a broader digital infrastructure stack —one that can support the movement of money and assets, programmable transactions, shared records, digital identity, and coordination across organizations.

The strategic question is therefore not which Web3 technologies a company should adopt. It is which parts of the stack, if any, improve a specific business process enough to justify the additional complexity.

This guide organizes the Web3 stack into eight practical layers, from blockchain networks and custody through smart contracts, tokenization, data, payments, and enterprise integration. Security, compliance, and governance cut across all of them.

Three principles shape the entire framework:

- ***Start with the business problem, not the technology.*** Blockchain infrastructure is most useful when it materially improves coordination, ownership, settlement, verification, or programmability.
- ***Different business goals require different stacks.*** A stablecoin payment system, tokenized asset, digital credential, and programmable B2B workflow will not require the same architecture.
- ***Web3 will usually be part of a hybrid system.*** Banks, enterprise software, APIs, custodians, compliance providers, and conventional databases will continue to operate alongside blockchain-based infrastructure.

The objective is not to maximize decentralization or put more activity on-chain.
It is to design the least complex architecture capable of creating the desired business outcome.

By the end of this guide, a reader should be able to understand the major components of the Web3 stack, recognize how they fit together, identify which layers a particular use case may require, and ask better questions before selecting technology or designing a pilot.

The central idea is simple: **You don't need every layer. You need the right ones.**

1. Why the Web3 Stack Matters

Business leaders do not need to become blockchain engineers.

But they increasingly do need to understand the infrastructure behind digital assets, stablecoins, tokenization, programmable transactions, and other emerging financial and commercial systems.

That infrastructure is often described as the Web3 stack. The term can make the technology sound more complicated than it needs to be.

At a practical level, the Web3 stack is simply the collection of technologies that allow organizations to:

- record and transfer digital assets;
- move money across blockchain networks;
- establish identity and permissions;
- execute business rules programmatically;
- connect blockchain systems to outside data;
- safeguard assets and credentials;
- integrate blockchain infrastructure with existing enterprise systems;
- and coordinate transactions across organizations.

No business needs every component.

The important strategic question is not: ***Which Web3 tools should we use?***

It is: ***What infrastructure does this particular business problem require?***

A company evaluating stablecoin payments will need a different stack from one tokenizing financial assets.

A company automating settlement between business partners may require a different architecture from one using blockchain-based credentials. And in many cases, the correct solution will combine blockchain infrastructure with conventional databases, banks, payment systems, APIs, cloud services, and enterprise software.

That is why understanding the stack matters. It helps organizations evaluate Web3 as infrastructure, rather than as a collection of disconnected technologies.

THE WEB3 STACK AT A GLANCE

The Web3 stack is a collection of infrastructure layers that work together to enable secure, scalable, and programmable digital business systems.

STRATEGIC TRUTH

No business needs every layer. The goal is to use the right infrastructure, in the right places, for the right business reason.

1. **Blockchain Networks & Settlement**
   The foundational networks where transactions are recorded and finalized.
2. **Wallets, Custody & Key Management**
   Infrastructure for controlling digital assets and authorizing transactions securely.
3. **Identity, Permissions & Access**
   Verifying who participants are and what they are allowed to do.
4. **Smart Contracts & Programmability**
   Business rules that execute automatically when conditions are met.
5. **Tokenization & Digital Asset Representation**
   Turning real-world value, rights, or assets into digitally transferable representations.
6. **Data, Oracles & Off-Chain Integration**
   Connecting blockchain systems to real-world data and external events.
7. **Stablecoins, Payments & Liquidity**
   Moving money and value through blockchain-based financial rails.
8. **Interoperability & Enterprise Integration**
   Connecting blockchain infrastructure with existing enterprise systems and networks.

CROSS-CUTTING-FOUNDATION

Security - Compliance - Governance

You don't need every layer.
You need the right ones.

2. The Web3 Stack Explained

There is no single universal Web3 stack.

Different applications assemble different technologies depending on what must be recorded, transferred, verified, programmed, protected, and integrated. For business purposes, it is useful to think about the stack in eight layers.

Layer 1: Blockchain Networks and Settlement

Where transactions are recorded and finalized

Blockchain networks provide the shared transaction infrastructure on which digital assets and programmable transactions operate.

They establish the rules for recording transactions, maintaining shared state, and determining when activity is considered final.

Different networks make different tradeoffs involving:

- **security;**
- **speed;**
- **transaction cost;**
- **decentralization;**
- **privacy;**
- **programmability;**
- **interoperability;**
- **and institutional control.**

For a business, the most important question is rarely which blockchain has the largest community or the most attention. It is: ***What kind of transaction environment does the use case require?***

Business considerations

Ask:

- Who needs access to the network?
- Who is permitted to validate or observe transactions?
- How quickly must transactions settle?
- What level of privacy is required?
- How predictable must transaction costs be?
- Does the application need to interact with other networks?
- Is a public, permissioned, private, or hybrid environment most appropriate?

Common use cases

- Digital asset issuance
- Stablecoin settlement
- Tokenized securities
- Multi-party recordkeeping
- Cross-border transactions
- Programmable financial workflows

Layer 2: Wallets, Custody, and Key Management

Who controls digital assets--and how

Blockchain systems rely on cryptographic keys to authorize activity.

That creates a new operational question:
Who controls the keys that control the asset?

For individuals, that function is often associated with a wallet.

For enterprises and institutions, the architecture can be much more sophisticated.

Organizations may rely on:

- **institutional custodians;**
- **embedded wallets;**
- **multi-signature authorization;**
- **multi-party computation;**
- **hardware security modules;**
- **role-based permissions;**
- **transaction policies;**
- **recovery procedures;**
- **and internal treasury controls.**

The business issue is therefore larger than “which wallet should we use?”
It is about control, authorization, security, and operational responsibility.

Business considerations

Ask:

- Who can initiate transactions?
- Who must approve them?
- Where are keys stored?
- What happens if credentials are lost or compromised?
- Does a regulated custodian need to hold the asset?
- How will treasury policies be enforced?
- How will wallet activity connect with accounting and internal controls?

Common use cases

- Corporate stablecoin holdings
- Digital asset custody
- Treasury operations
- Tokenized asset management
- Embedded customer wallets
- Institutional transaction authorization

Layer 3: Identity, Permissions, and Access

Who is allowed to participate

Open blockchain networks can allow addresses to transact without traditional account structures. Businesses often require something more.

They may need to know:

- whether a participant has completed KYC;
- whether an investor is eligible to hold an asset;
- whether an employee has authority to approve a transaction;
- whether a credential is valid;
- whether a user belongs to a particular group;
- or whether information can be shared with a specific counterparty.

This creates an identity and permissions layer. The objective is not necessarily to place personal information on a blockchain.

In many systems, blockchain infrastructure is used instead to verify **claims, credentials, permissions, or eligibility** while sensitive data remains elsewhere.

Business considerations

Ask:

- What must the organization know about a participant?
- What information can remain private?
- Which permissions must follow the user or asset?
- Can eligibility be verified without repeatedly sharing underlying data?
- Which regulatory requirements apply?

Common use cases

- Investor eligibility
- KYC credentials
- Business identity
- Employee authorization
- Customer access
- Digital credentials
- Permissioned asset transfers

Layer 4: Smart Contracts and Programmability

Where business rules become executable

Smart contracts are programs that execute on blockchain infrastructure. They can govern what happens when predefined conditions are met.

This can include:

- transferring assets;
- releasing payments;
- distributing revenue;
- enforcing transfer restrictions;
- updating ownership records;
- calculating fees;
- managing collateral;
- executing settlement;
- or controlling access.

The important capability is not simply automation. Traditional software already automates business processes.

The difference is that smart contracts can **allow multiple organizations and assets to interact through shared rules without every participant operating the same internal system.**

Business considerations

Ask:

- Which business rules can be defined clearly?
- Which events should trigger an action?
- What happens when something goes wrong?
- Who can modify the rules?
- Are manual overrides necessary?
- Has the contract been tested and audited?
- What legal agreement sits behind the code?

Common use cases

- Escrow
- Payment automation
- Asset servicing
- Revenue distribution
- Collateral management
- Conditional settlement
- Transfer Controls

Layer 5: Tokenization and Digital Asset Representation

How ownership, value, and rights become digitally transferrable.

Tokenization allows an asset, claim, entitlement, or right to be represented digitally on blockchain infrastructure.

The underlying item might be:

- money;
- a security;
- a fund interest;
- a commodity;
- real estate;
- intellectual property;
- a credential;
- a membership right;
- or another contractual claim.

Instead, tokenization can change the infrastructure through which the asset is:

- issued;
- recorded;
- transferred;
- settled;
- programmed;
- serviced;
- or integrated with other digital systems.

The token does not automatically create the asset's economic value.

Business considerations

Ask:

- What exactly does the token represent?
- What legal rights accompany it?
- Who maintains the authoritative ownership record?
- What transfer restrictions apply?
- How is the underlying asset connected to the token?
- Who performs custody and servicing?
- What happens if the token and legal record conflict?

Common use cases

- Tokenized funds
- Tokenized Treasuries
- Digital securities
- Private-market assets
- Loyalty and access rights
- Digital credentials
- Collateral

Layer 6: Data, Oracles, and Off-Chain Integration

How blockchain systems learn what happened outside the blockchain

Blockchains are good at verifying activity that occurs within their own environment.

Businesses operate largely outside it.

- Interest rates change.
- Invoices are issued.
- Shipments arrive.
- Assets change value.
- Customers make purchases.
- Contracts depend on external events.

For a smart contract to respond to those events, blockchain systems need reliable connections to outside information.

These connections can include:

- APIs;
- data providers;
- oracle networks;
- enterprise systems;
- payment systems;
- market-data feeds;
- identity databases;
- and Internet-of-Things devices.

This layer is often one of the most important—and least visible—parts of the architecture.

Business considerations

Ask:

- Which external data does the transaction depend on?
- Who supplies that data?
- What happens if it is inaccurate?
- Is one source sufficient?
- How quickly must information update?
- Which enterprise systems must exchange data with the blockchain?

Common use cases

- Asset pricing
- Collateral valuation
- Insurance
- Supply-chain verification
- Financial settlement
- Conditional payments
- Compliance checks

Layer 7: Stablecoins, Payments, and Liquidity
How money moves through blockchain-based systems

Stablecoins have become one of the most important bridges between traditional financial activity and blockchain infrastructure.

At the simplest level, they allow dollar-denominated or other fiat-linked value to move across blockchain networks.

But their significance goes beyond payment.

Once an organization can acquire, hold, transfer, account for, and redeem stablecoins, digital money can potentially interact with:

- treasury;
- supplier payments;
- customer payments;
- cross-border operations;
- settlement;
- smart contracts;
- tokenized assets;
- and liquidity management.

That can make stablecoins less like a new payment button and more like a financial operating layer.

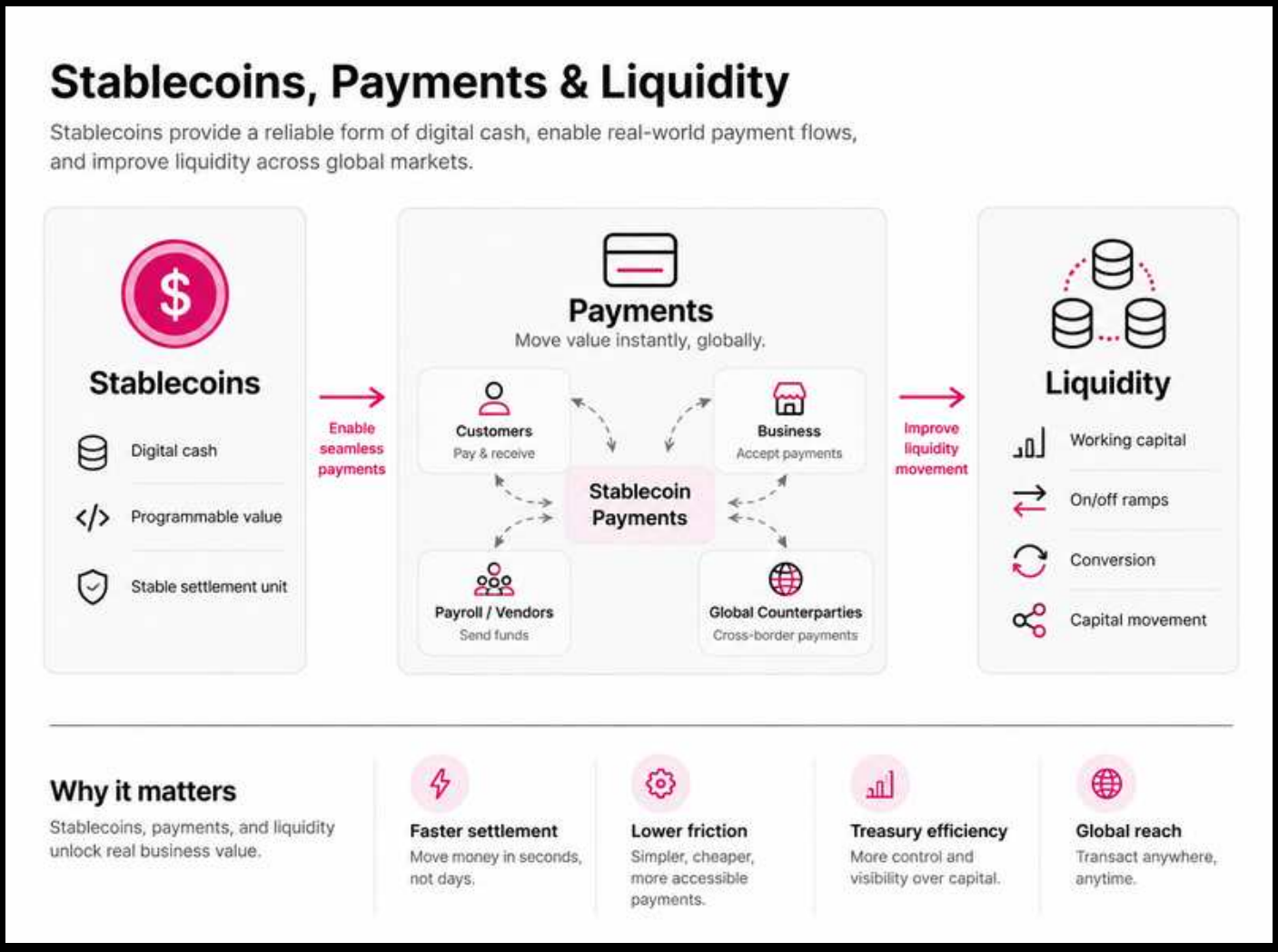
Business considerations

Ask:

- Why is a stablecoin preferable to the current payment method?
- Who issues and backs it?
- How is redemption handled?
- Which networks does it operate on?
- Where is liquidity available?
- How will it be accounted for?
- What compliance requirements apply?
- How will stablecoins interact with existing banks and payment providers?

Common use cases

- Cross-border payments
- Supplier payments
- Treasury transfers
- Digital asset settlement
- Programmable payments
- Marketplace payouts
- Tokenized asset transactions



Layer 8: Interoperability and Enterprise Integration

How blockchain infrastructure connects to everything else

Few businesses will operate entirely on-chain.

A blockchain application may need to interact with:

- banks;
- payment processors;
- ERP systems;
- accounting platforms;
- CRM systems;
- cloud databases;
- compliance providers;
- custodians;
- identity systems;
- other blockchains;
- and existing customer applications.

That makes interoperability one of the most important layers of the business stack.

A blockchain application that cannot integrate with the rest of the organization may simply create another silo.

Business considerations

Ask:

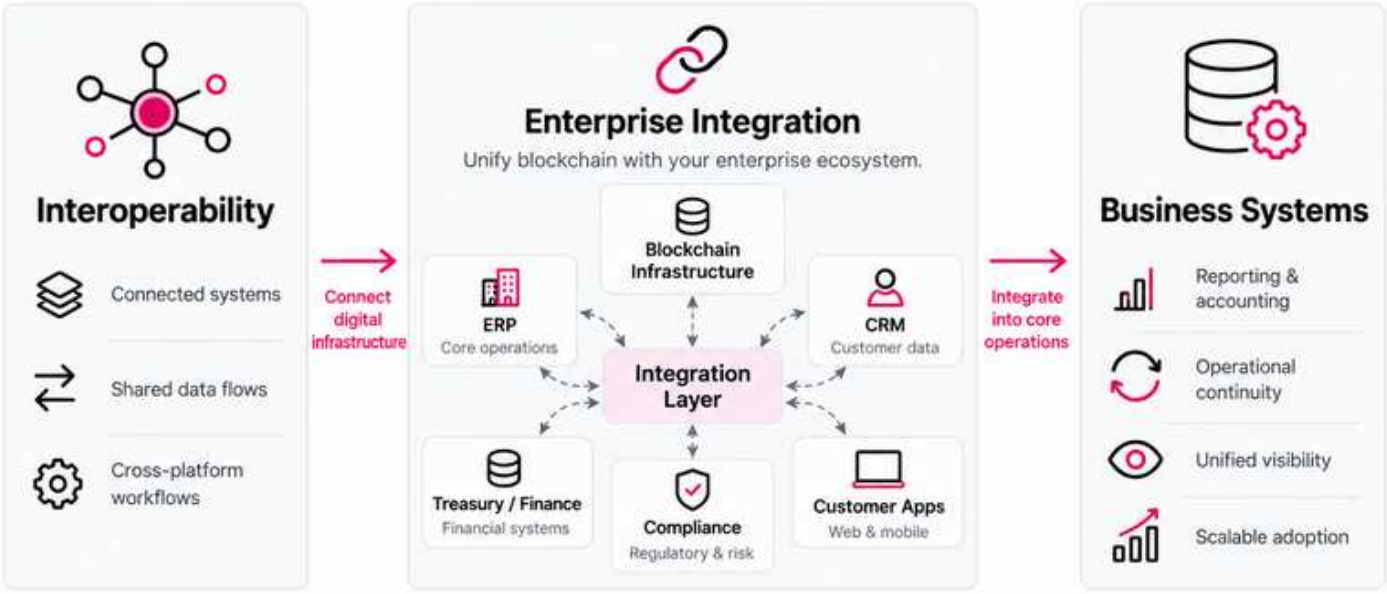
- Which existing systems must remain?
- Where will the authoritative record live?
- Which systems must exchange data?
- How will blockchain transactions appear in financial reporting?
- Can assets move between networks?
- How dependent is the architecture on one provider or ecosystem?
- What happens if a vendor or network changes?

Common use cases

- ERP integration
- Accounting
- Treasury management
- Cross-chain asset movement
- Bank/blockchain connectivity
- Compliance reporting
- Enterprise APIs

Interoperability & Enterprise Integration

Blockchain-based systems create the most value when they connect cleanly to the rest of the business.



Why it matters

Interoperability and integration turn blockchain into real business value.

No new silo
Connect blockchain to existing workflows.

Better coordination
Share data across teams and systems.

Operational visibility
Improve reporting, control, and traceability.

Scalable adoption
Use new infrastructure without replacing everything.

A Cross-Cutting Layer: Security, Compliance, and Governance

Security and compliance should not be treated as one more component added after the architecture is built.

They affect every layer.

A business must determine:

- who can transact;
- who controls assets;
- which transactions require approval;
- what information must be retained;
- which participants must be identified;
- what happens when assets are lost;
- how smart contracts can be changed;
- which jurisdiction governs a transaction;
- and who remains accountable when automated systems fail.

Good Web3 architecture therefore requires compliance-by-design, security-by-design, and governance-by-design.

The objective is not merely to make the technology work.
It is to make the system operationally trustworthy.

3. Building the Stack Around the Business Goal

Organizations should not begin by selecting technologies from each layer.

Begin with the business process. Then work backward. Here are several examples.

A. Stablecoin-Enabled Cross-Border Payments

Business goal

Reduce friction in moving money between customers, suppliers, subsidiaries, or international counterparties.

Likely stack components

Payment asset: Regulated stablecoin or tokenized deposit

Blockchain network: Network offering appropriate cost, speed, reliability, and liquidity

Wallet/custody: Institutional wallet infrastructure with corporate authorization controls

Identity/compliance: KYC, AML, sanctions screening, transaction monitoring

Integration: Treasury, ERP, accounting, and banking systems

Liquidity: On/off-ramp and redemption infrastructure

Why this architecture matters

The value may not come from one cheaper payment.

It may come from connecting payment, treasury, settlement, reconciliation, and liquidity through a more continuous financial infrastructure.

B. Tokenized Asset Issuance

Business goal

Improve how an asset is issued, distributed, transferred, settled, or serviced.

Likely stack components

Asset representation: Token structure tied to defined legal rights

Blockchain network: Settlement and ownership infrastructure

Smart contracts: Transfer controls, distributions, servicing, or compliance logic

Custody: Asset and key-control infrastructure

Identity: Investor eligibility and permissions

Data: Pricing, valuation, corporate actions, or underlying asset information

Payment layer: Stablecoins, tokenized deposits, or conventional cash settlement

Why this architecture matters

Tokenization creates the most value when it improves the asset lifecycle, not merely when it produces a digital token.

C. Programmable B2B Transactions

Business goal

Reduce manual coordination among organizations involved in the same commercial transaction.
Examples might include:

- milestone payments;
- escrow;
- supply-chain transactions;
- revenue sharing;
- insurance claims;
- or conditional settlement.

Likely stack components

Smart contracts: Shared transaction logic

Identity: Authorized participants

Data/oracles: Confirmation that contractual conditions occurred

Payment: Stablecoins or connected banking infrastructure

Enterprise integration: ERP, procurement, accounting, or workflow systems

Why this architecture matters

The opportunity is strongest when several parties currently maintain separate records and spend significant effort verifying, reconciling, or coordinating the same event.

D. Digital Identity and Credentials

Business goal

Allow participants to demonstrate identity, eligibility, qualifications, or rights without repeatedly exchanging the same underlying information.

Likely stack components

Identity infrastructure: Digital credentials or attestations

Wallet/access layer: Storage and presentation of credentials

Blockchain network: Verification or credential-status infrastructure

Privacy layer: Selective disclosure or privacy-preserving verification

Enterprise integration: Existing identity and access-management systems

Why this architecture matters

The objective is not to put someone's identity on a blockchain.
It is to make certain facts about identity or eligibility portable and verifiable.

E. Digital Loyalty, Membership, or Access

Business goal

Create portable or programmable customer rights, rewards, memberships, or access.

Likely stack components

Digital asset representation: Membership or entitlement

Wallet infrastructure: Ideally embedded in a familiar customer experience

Smart contracts: Rules governing access, rewards, or transfer

Identity: Customer permissions or credentials

Enterprise integration: CRM, loyalty, ecommerce, and analytics systems

Why this architecture matters

The Web3 component should improve the customer relationship in a way a conventional loyalty database cannot—not simply place existing points on a blockchain.

Different Stacks, Different Outcomes

There is no standard Web3 stack that every organization should adopt.

Different business problems require different combinations of networks, custody, identity, programmability, assets, data, payments, and integration.

The objective is not to use more of the stack—it is to select only the infrastructure necessary to produce the desired business outcome.

Architecture should follow the business problem—not the other way around.

4. How to Choose the Right Stack

Choosing infrastructure is a business architecture decision.

Before selecting a blockchain, wallet provider, smart-contract platform, custody solution, or token standard, evaluate these eight questions.

1. What problem are we solving?

Start with the existing friction.

Is the problem:

- settlement?
- reconciliation?
- ownership?
- access?
- verification?
- coordination?
- payment?
- liquidity?
- data portability?

If the problem cannot be defined clearly, the technology decision is premature.

2. Why does blockchain improve the architecture?

Always compare the proposed solution against the best conventional alternative.

Could the same result be achieved with:

- an API;
- database;
- existing payment provider;
- cloud service;
- workflow automation;
- or current financial infrastructure?

Blockchain should change something meaningful enough to justify the complexity.

3. Who needs to participate?

Web3 infrastructure becomes particularly interesting when multiple independent organizations need to coordinate around the same assets, records, or transactions.

Map:

- users;
- counterparties;
- banks;
- issuers;
- custodians;
- regulators;
- service providers;
- and internal teams.

Architecture follows participants.

4. What must be controlled?

Determine who can:

- issue;
- hold;
- transfer;
- approve;
- freeze;
- modify;
- redeem;
- or recover an asset.

Decentralized infrastructure does not eliminate governance.
It makes governance design more important.

5. What are the legal and compliance requirements?

Evaluate regulation before implementation.

Depending on the use case, this may include:

- securities regulation;
- payments law;
- KYC/AML;
- sanctions;
- custody;
- consumer protection;
- privacy;
- tax;
- accounting;
- licensing;
- and jurisdiction.

Compliance architecture can determine technical architecture.

6. What happens when something fails?

Plan for:

- lost credentials;
- incorrect transactions;
- smart-contract vulnerabilities;
- failed integrations;
- inaccurate data;
- network interruptions;
- vendor failure;
- and human error.

A production system needs more than a happy path.

7. How will this integrate with the existing business?

Ask where blockchain activity must appear in:

- accounting;
- treasury;
- CRM;
- compliance systems;
- internal reporting;
- data warehouses;
- and customer applications.

A system that cannot integrate operationally will rarely scale strategically.

8. How will we measure success?

Define the expected business result before implementation.

Relevant metrics might include:

- settlement time;
- transaction cost;
- working capital;
- reconciliation effort;
- error rate;
- fraud;
- conversion;
- customer retention;
- market access;
- operational headcount;
- or revenue.

The goal is not to prove blockchain works.

The goal is to determine whether the business works better because of it.

5. Six Infrastructure Trends to Watch

The Web3 stack continues to evolve, but several developments appear particularly important for business adoption.

1. Stablecoins Are Becoming Financial Infrastructure

Stablecoins are moving beyond crypto trading into payments, treasury, settlement, and cross-border financial operations.

The important shift is from thinking about stablecoins as an alternative payment method to considering how digital money can interact with software, assets, and business processes.

For businesses, the question is increasingly:

Where would continuously available programmable money change the operating model?

2. Tokenization Is Moving Toward Real Financial Assets

Tokenization is increasingly focused on assets whose economics already exist outside blockchain infrastructure.

That includes financial instruments, funds, Treasuries, private assets, collateral, and other real-world claims.

The strategic opportunity lies less in creating new token categories and more in improving:

- issuance;
- transfer;
- settlement;
- servicing;
- distribution;
- and collateral mobility.

3. Wallets Are Becoming Invisible Infrastructure

One of the largest barriers to Web3 adoption has been the requirement for users to understand wallets, seed phrases, gas, and blockchain transactions.

That is changing.

Embedded wallets, smart accounts, improved recovery, permission controls, and account abstraction are making blockchain interaction increasingly invisible to the end user.

For businesses, this means the Web3 experience may eventually look much more like conventional software.

That is probably a requirement for mainstream adoption.

4. Compliance Is Moving Into the Infrastructure

Web3 compliance is increasingly becoming a design problem rather than something layered on after a product launches.

Identity credentials, transaction monitoring, permissioned transfers, asset controls, and programmable compliance can increasingly be incorporated into system architecture.

The question will not simply be:
How do we make decentralized systems compliant?

It will also be:
Which compliance rules can the infrastructure enforce automatically?

5. Traditional Finance and Blockchain Infrastructure Are Converging

The future is unlikely to be entirely on-chain or entirely off-chain.

Businesses are increasingly likely to operate through hybrid architectures connecting:

- banks;
- stablecoin issuers;
- tokenization platforms;
- custodians;
- blockchain networks;
- payment providers;
- and conventional enterprise systems.

The critical infrastructure challenge will therefore be interoperability, not ideological purity.

6. Privacy and Verifiable Data Will Become More Important

Many business applications require both verification and confidentiality.

Organizations may need to prove:

- identity;
- eligibility;
- compliance;
- ownership;
- transaction status;
- or data integrity

without exposing all underlying information publicly.

Privacy-preserving technologies, digital credentials, and verifiable computation could therefore become increasingly important as institutional adoption expands.

6. A Practical Way to Think About the Stack

It is easy to look at eight layers and assume that Web3 infrastructure is inherently complicated.

That is the wrong conclusion. Most businesses will need only a subset.

Start with five questions:

What business outcome improves?

- Cost?
- Speed?
- Access?
- Revenue?
- Risk?
- Customer experience?

What are we trying to move?

- Money?
- Assets?
- Data?
- Rights?
- Credentials?

What must remain connected?

- Banks?
- ERP?
- CRM?
- Custody?
- Accounting?
- Existing applications?

What needs to be programmed?

- Payments?
- Permissions?
- Settlement?
- Ownership?
- Compliance?

Who needs to coordinate?

- One organization or several independent parties?

Those answers reveal the stack. Not the other way around.

From Infrastructure to Strategy

Understanding the Web3 stack is useful only if it leads to better decisions.
The objective is not to adopt more technology, move more activity on-chain, or build the most decentralized system possible.

It is to identify where new digital infrastructure can improve a real business process —and use only the components necessary to achieve that outcome.

Start with the operating model.

- Where does value move?
- Where does ownership change?
- Where do organizations reconcile separate records?
- Where does verification create friction?
- Where could shared rules reduce coordination?
- Where could programmable money or assets create a better process?

Those questions should determine the architecture. Not the other way around.

The next step

Before selecting technologies or providers:

- Evaluate the opportunity.
- Determine whether blockchain-based infrastructure offers a meaningful advantage over the strongest conventional alternative.
- Map the required stack.
- Identify the networks, custody, identity, programmability, asset, data, payment, and integration capabilities the use case actually requires.
- Compare architecture options.
- Consider conventional, blockchain-based, and hybrid approaches—and determine what should be built, bought, or provided by partners.
- Test before scaling.
- Design a contained pilot around a measurable business hypothesis rather than simply proving that the technology works.

The right infrastructure, in the right place, for the right business reason

Argot helps organizations evaluate and implement emerging digital infrastructure—from blockchain strategy and stablecoins to tokenization, smart contracts, and enterprise integration.

Explore Argot Strategy Services → argotagency.io

ARGOT