

Definition of a Solution

Instructions for Describing a Solution



Author: Danilo Assmann

Status: in work

Version: 1.0.3 (build 6)

Date: 2026-10-07

Copyright: © 2026. This work is openly licensed via [CC 4.0 BY SA](https://creativecommons.org/licenses/by-sa/4.0/)



OrgIQ Manifest: Passion for People builds Systems for Success

OrgIQ—Organizational Intelligence (Quotient)—is a framework that redefines how we see organizations. At its heart lies a simple but profound shift: to think from the perspective of the individual. Every person experiences their own reality, shaped by their network of relationships. Each perspective is unique. There is no single right or wrong.

An organization is the superposition of all these *Networks*. This is complexity. We can never fully model or control it—but we can give it direction and purpose. Intelligent organizations embrace this complexity rather than ignore or oversimplify it.

When we focus on human complexity, the range of possible solutions expands dramatically. Business practices and structures can be brought into harmony with the natural dynamics of human relationships, psychology, and social interaction.

Our manifesto is rooted in lived experience. We have seen the transformative power of genuine connection, individuality, and purpose-driven leadership. OrgIQ helps create environments where empathy, understanding, and mutual respect thrive—where people feel seen, heard, and valued. This sense of emotional safety is the foundation on which individuals and groups reach their full potential.

We believe in value over control. Control breaks perspective, creates overhead, and breeds mistrust. Relationships and trust, on the other hand, unleash extraordinary productivity and fulfillment. Purpose is not measured by words, but by results.

Join us in redefining organizational excellence. Let's build intelligent systems, embrace complexity, and make space for purpose and joy. **Welcome to OrgIQ—where the true potential of your people becomes the strength of your organization.**

Content

Solution describing a service and its architecture.....	4
What this document is for.....	4
You choose the depth.....	4
A Mini-version the Solution card.....	5
Why and how.....	5
Mini-check can we work with this.....	6
Sentence templates for the mini-card.....	6
B Optional extensions where they help.....	7
0 Compare described lived and desired architecture.....	7
1 Extend need contribution and boundaries.....	7
2 Extend character human conditions and priorities.....	8
Chameleon or a clear character of your own.....	8
3 Extend internal architecture and actual agency.....	8
Optional three views for more precise design.....	8
4 Extend functional and non-functional requirements.....	10
5 Design interfaces and intelligent gateways.....	10
Describe a gateway as an architectural element.....	10
6 Examine use effects side effects and the four C.....	11
7 Extend independence effort and downstream costs.....	11
8 Extend Learn and Adapt and lifecycle.....	12
C Worked example IT service for different users.....	13
Mini-card S-IT-01 Digital working environments.....	13
What is shared and what varies.....	13
Gateway contributions and intelligence.....	14
Completing your own work.....	14

Solution describing a service and its architecture

Organization / area: [...]

Participants / date: [...]

What this document is for

Describe a product, service or internal supporting contribution as a connected unit of benefit. Start with the required effect, explain realization, and make conditions, connections and learning visible.

Prefer beginning with outward services. Then use the same template for required supporting services such as IT, infrastructure, people services or quality assurance. A department is not automatically a Solution: it may carry several services, and several participants may jointly provide one service.

Overall architecture belongs in the [Organisation Guide](#). Here you describe one Solution's internal architecture and its connections to the whole. Reference shared capabilities. Larger groupings can provide an overview without creating another command level.

Atomic means that, for your intended use, the unit has a complete contribution loop and accountable connections. It does not mean technically indivisible, partner-free or one team. Further decomposition requires distinct contributions and connections for the resulting units.

You choose the depth

The **mini-version** is independently usable. It records minimum orientation, connections and the next check. Then choose individual **extensions**. You do not have to complete the entire document to begin.

More architecture work can reduce later friction; more text does not guarantee it. A shorter description can suffice where relationships and practices already work. Effects, understanding and the consequences of actual gaps determine quality.

Practical start: Complete the mini-card together, walk through a normal case and a disruption, and select the most important open question. Extend there. Depth remains your decision; agreed obligations and harm boundaries apply regardless of documentation.

When time is scarce: One sentence per field is enough for an initial draft. "Unknown" is acceptable. Add only the most important open question, who handles its consequences and the next review point. Other modules are optional. A short card means little description, not necessarily little existing architecture; a long card does not prove functioning collaboration.

Status for answers: **assumption** · **observation** · **agreed** · **evidenced** · **contradicted** · **unknown**. Description and evidence are different.

A Mini-version the Solution card

Why and how

A short card connects need and benefit with the offer, working practices, people and learning. Initially write one to three sentences per field. Reference existing agreements rather than copying them. “Unknown” is useful when it leads to a next check.

ID / name / version / date: [...]

Connection to overall architecture / other Solutions: [...]

Field	Guidance	Your description
1. Need and purpose	Who needs what? What effect do we enable? What wider contribution does it support? For internal services, which named Solutions connect it to external need?	[...]
2. Scope and boundary	What do we own? What do we require from others? What explicitly lies outside our responsibility?	[...]
3. Offer and use	What do recipients receive? How should they use it, and what effect is expected?	[...]
4. Character and people	Which few properties define the work? What conditions and agency do participants need?	[...]
5. Realization	How is the service produced? Who can decide what? What resources or partners are required?	[...]
6. Quality conditions	What functional criteria and non-functional boundaries apply? What mechanisms support them?	[...]
7. Connections	What services do we need, from whom and why? Where is meaning sufficiently shared for an interface, and where is an intelligent gateway needed? Name recipient, provider and connection ID.	[...]
8. Review	How do we examine use, expected effects and unexpected consequences? What is evidenced or open?	[...]
9. Learn & Adapt	Who can change things? What support remains necessary? When do we review, change or transfer the form?	[...]

Our next step: [...]

Key gap / accountable connection / next check: [...]

Chosen extension or deliberately none yet: [...]

Represented perspectives / missing view / unresolved dissent: [...]

Key compensating work for an unclear prerequisite / who carries it / next review: [...]

You need not view the same service from the same perspective. In a real case, check whether provider and recipient understand the contribution, conditions and connections sufficiently consistently. This is the shared basis for action. Their different working worlds remain visible; design a gateway where translation is necessary.

Mini-check can we work with this

- ▶ Can a recipient understand the service and its boundary?
- ▶ Are necessary prerequisites and connections named?
- ▶ Is its direct, supporting, maintaining or future contribution to external need traceable? Is an uncertain future hypothesis explicitly linked to learning?
- ▶ Can participants recognize their actual decision rights?
- ▶ Are essential human and technical quality conditions considered?
- ▶ Is there a path from actual effects to change?

An open question does not always prevent starting. Record what must be resolved before starting and what a bounded trial can establish. Missing prerequisites remain visible.

Sentence templates for the mini-card

Subject	Sentence	Example
Need	"[Users] need [capability] because [situation/consequence]."	"Development teams need readily available test environments because waiting blocks experiments."
Purpose	"We enable [users] to [effect], contributing to [benefit]."	"We enable suitable experiments in protected environments so teams can learn sooner."
Offer	"We provide [service] for [use situation] within [scope]."	"We provide a temporary test environment for agreed data and applications."
Boundary	"We own [...]; [...] remains with [...]."	"We own the environment; evaluation of the experiment remains with development."
Property	"In [situation], [property] must hold; enabled by [mechanism], checked through [evidence]."	"During disruption, next steps must be clear; a named contact and status route support this."
Learning hypothesis	"If we change [...], we expect [effect]; we check [...] and watch for [side effect]."	"A standard request should reduce waiting; check lead time and additional applicant work."
Interface	"[A] provides [B] with [contribution] under [conditions]; for [deviation], use [route]."	"The team provides data class and test needs; unclear data classification is resolved together before provision."
Gateway	"We translate between [logic A] and [logic B] to enable [shared contribution]."	"We translate a successful experiment into maintainable operating conditions before it becomes a production service."

For specific functional expectations, use a user story: "As [user], I want [capability] so that [benefit]." Add scope, conditions and evidence. A user story alone describes neither overall purpose nor complete architecture.

B Optional extensions where they help

0 Compare described lived and desired architecture

Why? Good results can arise from reliable mechanisms or recurring personal compensation. Description should reveal actual work. Required effects and relevant conditions are the reference, not a generic ideal organisation.

What to do: Choose a real normal case or recurring disruption. Trace use, contribution, decisions, handovers and feedback with the affected perspectives. Unknown current-state statements remain unknown. Do not record a proposal as an existing rule.

Subject / required effect	Described / existing commitment	Lived / case and evidence	Desired / relevant condition	Gap / benefits and burdens / explanation and alternative hypothesis
[...]	[...]	[...]	[...]	[...]

Questions: Where do people wait, ask again, translate or improvise? Who can actually decide or override? Who rescues the result? What costs and burdens arise? What do participants experience and how do they respond? What existing mechanism is helpful and worth retaining?

Hypothetical example: A team is meant to decide within its budget. Decisions are subsequently reversed without recognizable criteria. The team seeks advance approvals; management perceives little initiative. The hypothesis is that precaution is a reasonable response to unreliable authority. Alternatives such as competence gaps or unclear risks remain to be examined. A change must establish actual boundaries and intervention criteria; check effects through decisions, compensating work and quality. This is an illustrative case, not a finding about a real company or person.

Sentence: “The agreement is [...]. In practice [...]. For [effect], we need [...]. We suspect [...]; alternatively [...]. We change [...] and check [...], including side effects.”

Ideas and decisions: Name criteria and scope. Rejection identifies unmet criteria, missing evidence or prioritization. Where useful, provide a concrete revision and reassessment route. Changes, exceptions and personal preferences remain visible. Shared terms are insufficient; use examples to check reasonably consistent application.

Result: An effect-based gap, testable explanation and bounded change. An insight begins investigation; it does not prove a cause. Fuller facilitation is in the [Perspectives Module](#).

1 Extend need contribution and boundaries

Why? A named product can serve several needs. A department can provide several services. A useful boundary follows the intended benefit.

What to do: Observe real use or trace a concrete need. Compare realizations. Determine whether a service needs variants or several distinct Solutions.

Fill in: Beneficiaries [...] obstacle [...] effect [...] alternatives [...] own contribution [...] prerequisites [...] deliberate limits [...].

Boundary rule: Variants fit when the same core contribution serves different use conditions. Separate Solutions are useful when effect, responsibility, mechanism or evidence differ substantially and form distinct contribution loops. Shared capabilities remain connected.

Example: Identity and access rights may be shared. An experimental environment and an ongoing production service can build on them with different commitments and evidence. One technical platform can therefore support several Solutions.

Result: Justified scope and connection to overall architecture.

2 Extend character human conditions and priorities

Why? Recipients and providers can have different working logic. Fit does not require sameness.

What to do: Describe your own character, choosing at most three defining properties. Examine shared organizational boundaries, local minimum conditions and deliberate trade-offs.

Questions: What matters to recipients and participants? Which differences help? Where are trust, knowledge, energy or decision space missing? What is promised versus experienced? What collaboration can be sustained?

Fill in: Our character [...] is visible through [...]. For users with [...], we offer [...]. Participants need [...]. We prioritize [...] and accept [...] without compromising [...].

Chameleon or a clear character of your own

A chameleon can express adaptability. It must not mean changing identity, boundaries and promises for every wish. An adaptable system also needs a common core.

IT example: A reliable enabler provides safe, understandable digital working options. Research needs fast bounded experiments; production needs stable operation and planned changes. IT develops suitable offers around a shared core rather than becoming both worlds in their entirety.

Animal images: Octopus for distributed perception and flexible responses; beaver for reliable shared infrastructure; cheetah for short experiments; tortoise for stable protected routines. Translate images into authority, information, checks, capacity, recovery and handovers. They do not classify people.

3 Extend internal architecture and actual agency

Why? The service needs sustainable work and decisions rather than continual invisible rescue.

What to do: Describe processes, patterns, roles, knowledge, resources and relationships for normal operation, disruption, learning and transition.

Capability / step	Input → result	Role / actual authority	Resources / knowledge	Handover / disruption
[...]	[...]	[...]	[...]	[...]

Protective architecture check: Who always approves? Who can reverse local decisions? What legitimate concern is addressed? Where are competence or resources missing? What happens after unwelcome feedback or a successful result without leadership intervention?

Procedure: Observation [...] possible function [...] alternative design [...] actually granted authority [...] bounded trial [...]. Personal protective functions remain hypotheses unless clarified voluntarily. Technical, external or historically evolved causes are also possible.

Result: Responsibility matches agency. Autonomy is actually supported rather than merely promised.

Optional three views for more precise design

Why? A process can be described without its promised result, or a result promised without the conditions needed to achieve it. Three views separate and connect these questions and their checks.

Cleanroom box structures distinguish external behavior with input history (Black Box), relevant retained states and transitions (State Box), and implementing procedures that may use further boxes (Clear Box). Source: [SEI Cleanroom Software Engineering Reference section 2.2.](#)

Our organizational adaptation: Use the distinction as design support. It neither completely specifies nor mathematically proves living organizational behavior. Context, relationships and participants' own decisions remain influential. Alongside commitments, record hypotheses, observations and uncertainty.

View	Main question	Record
Service promise / Black Box	What can recipients expect?	Need, scope, inputs and relevant history, promised service, conditions, expected use/effect and harm boundaries
Conditions and state / State Box	What must inform an appropriate response?	Capacity, commitments, approvals, knowledge, dependencies, responses under different conditions and unknown influences
Realization / Clear Box	How do we enable this service?	Processes, patterns, roles, authority, resources, relationships, sub-Solutions, interfaces, intelligent gateways and Learn & Adapt

These are views of one element, not new departments. Connect required sub-Solutions' service views to their own descriptions using shared IDs. Gateways can also be described in these views.

Start with three sentences; add a table only if needed:

- ▶ “For [recipients], in [situation/prerequisites], we provide [service] to enable [effect].”
- ▶ “Our response depends on [conditions]; with [deviation], we agree [alternative/clarification].”
- ▶ “We realize this through [mechanisms/roles] and require [other Solutions' contributions].”

Hypothetical IT environment: Development receives a usable temporary environment for approved data within an agreed timeframe, enabling earlier experiments. Development owns experiment quality. IT can commit with available capacity and clear data classification. Unknown classification requires joint clarification. Capacity shortage requires a new agreement, not an impossible promise. Retaliation connects provision, approval, operation and removal through a gateway between research and operations.

Request	Relevant condition	Agreed response	Who can act?	Change / check
Test environment	Classification clear, capacity available	Provision within agreed timeframe	Provision role	Usable environment, capacity commitment and use check
Same request	Classification unknown	Gateway clarifies use and data requirements with both sides	Gateway with mandate	Shared understanding or explicitly unresolved conflict
Same request	Capacity committed	Agree timing or alternative	Roles with prioritization authority	Sustainable commitment instead of hidden overload
[...]	[...]	[...]	[...]	[...]

Describe only states needed for reliable decisions. Trust, exhaustion and fear are not arbitrary measurable switches. Use voluntary feedback, observable conditions and consequences; do not store private emotional profiles as operating state.

Refinement check: Can realization support commitments under the stated conditions? Are functional and non-functional requirements, including human conditions, covered? Can people act on deviations? Examine normal operation, disruption and transition. Verify implementation against agreements; validate actual use, benefit and consequences with recipients. Coherent design does not replace operational observation.

Result: Clear external description, visible decision conditions and aligned internal architecture. Depth remains your choice; unknown prerequisites stay visible.

4 Extend functional and non-functional requirements

Why? An offer can fulfill its function and still be unusable, exhausting or unreliable.

What to do: Connect requirements, mechanisms and evidence. Adopt higher-level boundaries and add local conditions. Some non-functional properties emerge through interaction rather than identical requirements for every part.

ID / requirement	Origin / scope	Target or minimum	Mechanism	Responsibility	Check / trade-off
[...]	[...]	[...]	[...]	[...]	[...]

Search space: Trust, openness, truthfulness, relationships, energy, safety, agency, learning, speed, quality, availability, robustness, economic viability, scalability, maintainability, knowledge transfer, handover and resource effects.

Priority: Boundary / character-defining / sufficiently good / later / irrelevant. Give reasons. Choose at most three next changes based on affected benefit, harm, dependencies and feasibility. Important uncertain relationships may first require observation.

Example: Learning needs time, feedback and change authority; a monthly meeting alone does not demonstrate it. Energy needs capacity and the ability to renegotiate excessive commitments; a workload questionnaire alone adds no capacity.

Result: Functional and non-functional conditions are supported or visible as gaps with accountable connections.

5 Design interfaces and intelligent gateways

Why? Two sides can use the same words while expecting different contributions. Handover alone then leaves misunderstanding unresolved.

What to do: Compare language, meaning, pace, risks and working logic for the actual connection. Explain whether shared meaning suffices or translation needs intelligence.

Connection	Condition	Required capability
Interface	Sufficiently similar worlds; terms, promises and conditions understood consistently	Clear handover and rule-based execution; the interface may be simple
Gateway	Differences cause misunderstanding without translation	Perceive, understand, translate both ways, judge, check understanding and learn

These are organizational working definitions. Technical interfaces may exist in either case. A common form alone does not create shared meaning. Walk through an actual exchange and let both sides explain the words they use, what they need and the consequences of their expectations before choosing the connection.

Connection	World A / meaning	World B / meaning	Common ground / difference	Interface or gateway / rationale
[...]	[...]	[...]	[...]	[...]

Describe a gateway as an architectural element

- ▶ ID / worlds: [...]
- ▶ Purpose / translation contribution: [...]
- ▶ Inputs / results: [...]
- ▶ What must be understood and translated: [...]
- ▶ Provider / generalists / specialist access: [...]
- ▶ Actual authority / resources / capacity: [...]

B Optional extensions where they help

- ▶ **Conflict and escalation route:** [...]
- ▶ **Evidence:** Can both sides explain meaning and commitments? Does required benefit arise? What misunderstandings remain? [...]
- ▶ **System IQ:** What can the gateway do itself, and what support does it need for new cases? [...]
- ▶ **Learn & Adapt:** How does feedback change translation and, where necessary, connected Solutions? [...]

Generalists help connect perspectives but need specialist access, time, trust and authority. The capability may reside in people, shared practices and tools. Stable translation can sometimes be automated; new or contradictory situations still require suitable judgment.

If a gateway has its own complete contribution loop, use a mini-Solution card. Otherwise retain it as an accountable architectural element inside a Solution. In either case make its contribution and intelligence visible.

Examples: Standard access with mutually understood data can be an interface. “Fast” may mean time until the next experiment in research but recovery time during agreed production operation. A gateway clarifies actual need and translates it into service commitments. Transition from experiment to production also requires agreement on maintenance, data, responsibility and capacity.

Limits: Translation does not resolve genuine goal conflicts itself. It makes them understandable and enables authorised decisions. A gateway must not become an irreplaceable person's permanent bottleneck; include knowledge, cover and feedback.

Actual communication path: Can connected roles perform exchange, change and conflict resolution? Do they have access, time and authority? A drawn connection is insufficient. Describe needed and actual connections; design deliberately rather than making everyone communicate with everyone.

Result: Every important connection is justified as interface or gateway. Gateway capabilities, responsibility, effort and learning are visible.

6 Examine use effects side effects and the four C

Why? Delivery is not use, and use does not establish the intended effect. Good intentions do not demonstrate benefit.

What to do: Trace real use, required effects and unexpected consequences. Connect evidence to the relevant level and scope. Examine affected parties as well as direct recipients.

Question / level	Expected observation	Evidence / method / timeframe	Responsibility	Current status / next check
[...]	[...]	[...]	[...]	[...]

Check functional performance, human conditions and connections. Use observations, recipient feedback and relevant data. Do not replace a missing effect with activity counts. After changing an offer or pattern, review affected metrics; evidence for unchanged conditions may remain useful.

Verification: Do we meet agreed requirements? **Validation:** Is the offer suitable for the needed benefit in its actual context? Examine interaction as well as individual components.

Four C: Completeness covers relevant contributions and conditions; Closure covers necessary effect, handover and learning paths; Correctness means reliable statements and evidence; Coherence means purpose, components, relationships and behavior fit together. Rate **met within scope / gap / unknown**, with reasons.

Result: An honest evidence position and next decision. A completed description is not proof of success.

7 Extend independence effort and downstream costs

Why? Good outcomes can arise from sound architecture or continuous compensation. The distinction determines dependencies and costs.

What to do: Examine orientation, perception, decisions/operation, relationships, learning and renewal individually. Use the six OrgIQ stages as a working model:

B Optional extensions where they help

1. **External dependence:** Instructions and repairs must be supplied.
2. **Independent operation:** Known cases work within the given framework.
3. **Adaptive agency:** Deviations are recognized and appropriately handled.
4. **Learning and repair:** Causes, assumptions and architecture are changed.
5. **Distributed intelligence:** Sufficiently connected participants carry perception, judgment and learning.
6. **Purpose renewal and interdependence:** Relevance is reviewed and form responsibly renewed, transferred or ended.

Profile: Capability [...] observed independence [...] evidence [...] support still needed [...] next development step [...]. Do not average these into a simple overall grade. This is not a personality test or scientifically validated score.

Separate effort: Necessary service / deliberate collaboration and development / avoidable architectural compensation. External support is not automatically a deficit. Documentation work is not automatically beneficial debt reduction.

Cost register: Gap [...] frequency [...] effort/range [...] consequences [...] possible repair [...] choice to extend or consciously accept [...] next review [...]. Avoid double counting. IQ alone does not determine financial cost.

Distinguish necessary communication and learning from **substitute steering**: repeated personal intervention that compensates for missing shared direction, agreements, authority or feedback. Establish whether a real gap exists or only documentation is missing. Not every open question is avoidable architectural debt.

Decision support: Compare additional design and implementation effort with expected reduction of recurring compensation and risk over the same period. State assumptions and examine actual relief. Hypothetically, repeated clarification consumes two hours weekly. Twelve hours spent clarifying a connection could pay off after twelve weeks if one hour per week is subsequently saved. This is a testable expectation, not a savings guarantee.

For disputed perspectives or human feedback loops, use the [optional Perspectives Module](#). One relevant case suffices; a complete profile is unnecessary.

Result: An informed depth decision with visible work and risks arising from unresolved gaps.

8 Extend Learn and Adapt and lifecycle

Why? A Solution must perceive its effects and respond to changing needs instead of making its current form an end in itself.

What to do: Define signal, shared interpretation, hypothesis, authority, experiment and effect review. After changes, review affected requirements and connections; evidence for stable conditions can remain.

Fill in: Signal [...] hypothesis [...] bounded change [...] authority/resources [...] review of effects and harm [...] incorporation/revision [...].

Lifecycle: What effect can be achieved, and what must be maintained? When do we examine relevance and alternatives? What obligations remain during transfer? Where could people's and capabilities' distinctiveness create greater benefit for themselves and the wider system?

Result: A learning loop and responsible transition. Purpose fulfillment percentages need defined conditions; an ongoing effect cannot simply be declared finished.

C Worked example IT service for different users

All statements below are hypothetical design decisions, not findings about an actual organization or established outcomes.

Mini-card S-IT-01 Digital working environments

Field	Example
Need and purpose	Teams need suitable digital environments. We enable usable, responsibly operable environments so experiments and ongoing services do not fail through unclear IT support.
Scope and boundary	IT owns agreed environment, access and support. Users own professional use and supply data class and operating needs. Professional decisions and arbitrary application development do not automatically belong to this service.
Offer and use	Two agreed profiles: a temporary experimental environment and a supported environment for recurring work. Production deployment needs separate clarification; “test” or “pilot” implies no production commitment. Users use the selected profile for the stated work.
Character and people	A reliable enabler: understandable, reliable and adaptable. Users have clear options; IT can expose risks and capacity limits without punishment. Safe work and honest information apply to both profiles.
Realization	Clarify need/data → select profile → provide environment → check access/use → support → learn. Qualified roles have agreed authority, cover and capacity.
Quality conditions	Data and access rules are explicit. Promises, availability and support are defined per profile. Feedback, knowledge transfer and capacity review prevent hidden overload.
Connections	Shared standard information uses interfaces. G-IT-F connects research and IT; G-IT-P connects production and IT; a transition gateway connects experimentation and ongoing operation.
Review	Observe use, time to usable environment, fitness for work, disruption, support effort and strain. Agree baseline and target ranges before the pilot; outcomes are not established yet.
Learn & Adapt	Feedback reaches IT and users together. After the pilot they prioritise one change. Before an experimental environment expires, decide whether to stop, extend or transfer it into a suitable operating Solution. Reassess effects and burdens after changes.

What is shared and what varies

Aspect	Shared core	Experimental profile	Recurring work profile
Purpose	Enable usable digital work	Earlier suitable experiments	Reliable ongoing work
Character	Reliability, clarity and adaptability	Short bounded learning cycles	Stable operation and planned change
Conditions	Safe data/access and honest commitments	Agreed experiment limits	Agreed availability and recovery

Aspect	Shared core	Experimental profile	Recurring work profile
Working practices	Clear expectations and truthful feedback	More iteration and adaptation	More predictability and repeatability
Resources	Visible capacity and competence	Limited duration and learning budget	Agreed ongoing operating effort
Evidence	Actual use and required effect	Experiment suitability and learning	Ongoing suitability and fulfilled commitments

Gateway contributions and intelligence

Gateway	Different worlds	Contribution	Capability and responsibility
G-IT-F Research ↔ IT	Open learning question and iteration ↔ data, technology and capacity	Translate experimental need into a suitable bounded environment and clear promises	Generalists clarify unknowns with both sides, check use and adapt; specialist decisions remain with authorised roles
G-IT-P Production ↔ IT	Continuity and recovery ↔ technical services and changes	Translate operating consequences into appropriate availability, support and change conditions	Connected roles understand process consequences and IT dependencies, check commitments and support conflict decisions
G-IT-T Experiment ↔ ongoing operation	Learning trial ↔ reliable continuing service	Translate learning into an accountable operating model	Clarify resources, maintenance, data and responsibility; an approval checkbox is insufficient

Gateway IQ and review: Unknown before trial. Observe whether translation works only for familiar cases, handles deviations independently and changes offers through experience. Examine understanding, failed promises, rework and use benefit, not merely contact counts.

Where does an interface suffice? Shared identity, standard information and a known support route where meaning is shared.

Where is a gateway needed? Transition into ongoing operation or differing risks, language and time horizons. Participants clarify the necessary service architecture; approval alone does not translate.

When to split the Solution? A continuing production service with its own operating model and commitments may become a separate Solution. Identity, platform and knowledge remain shared referenced capabilities. Do not reinvent the entire IT service for every recipient.

Four C conclusion: Profiles, boundaries, learning and gateway contributions are described. Confirm authority, resources, data conditions and commitments before the pilot. The effect hypothesis remains to be tested. Connect different characters through suitable offers without arbitrarily changing core values.

Next step: One experimental team and one team with recurring work each examine a real use case with IT. Determine necessary information and transitions, then extend the key gap rather than filling every field pre-emptively.

Completing your own work

We chose: Mini-version / extensions [...].

Current description and evidence: [...].

Connection to overall architecture: [...].

Deliberately open questions and consequences: [...].

Next decision / participants / review: [...].

This is an OrgIQ working model for piloting. Depth remains your choice. Actual effects, human sustainability, clear responsibility and an actionable learning loop remain the reference.

Optional appendix: [Architecture principles and their origins](#).

Background: [OrgIQ](#), [NASA Validation](#), [Sociocracy 3.0 Domains](#). Interface and gateway are deliberately used as organisational working terms here.