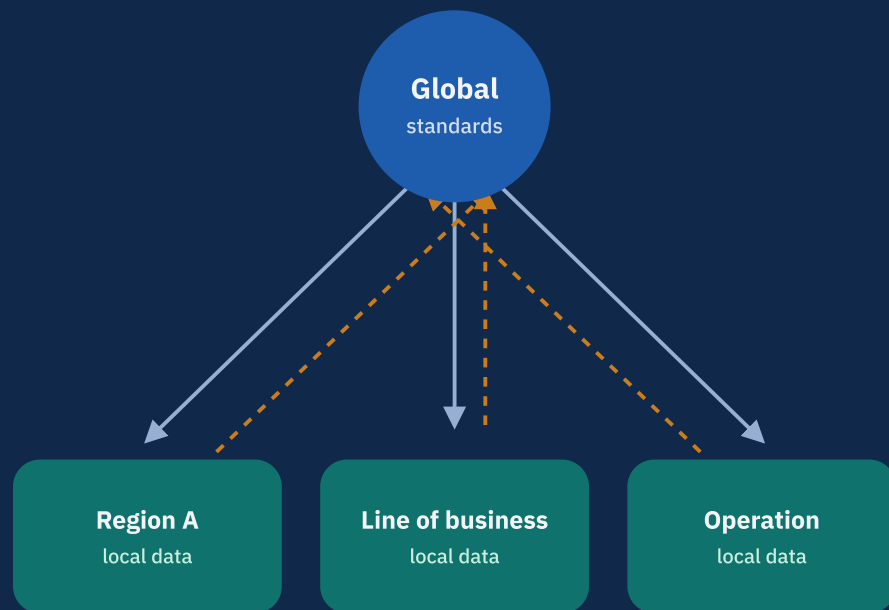


WHITEPAPER

Federated resilience: **one program, many jurisdictions**

How a single resilience program can respect data-residency rules, autonomous lines of business and independent operations, without splitting into disconnected tools.



standards and shared entities flow down · summaries flow up

EXECUTIVE SUMMARY

A resilience program has to be one program, even when the data cannot be in one place

Operational resilience is judged across the whole organization: boards and regulators ask whether critical services can stay within tolerance, wherever they run. At the same time, the data behind those answers is increasingly bound to where it was created. Privacy and residency laws, sovereign-cloud rules, and the independence of business units and government operations all push the same way: **keep the data local**.

A single central database cannot meet both demands. A set of disconnected local tools cannot either. This paper describes a third approach, **federation**: each jurisdiction or unit runs its own complete resilience platform; common standards and shared entities are owned once and flow down; and only agreed summaries flow up to give the enterprise view.

The paper explains the model, the design choices behind it, and the questions any organization should ask before choosing a platform for a multi-jurisdiction program.

THE PROBLEM

Three reasons one database stops working

Data residency

A multinational with operations across the Americas, Europe, the Middle East and Africa has to keep employee, site and incident data in the country where the law places it. Some regulators also require in-country hosting for critical systems.

Autonomous lines of business

Groups with near-independent businesses (insurance and asset management, manufacturing and retail) let each run its own continuity program, yet the group still sets the method and answers to the board for all of them.

Independent operations, common functions

Public agencies run many operations on shared HR, IT, facilities and finance. Each operation plans on its own, but they all depend on the same common functions, and a failure there hits all of them.

Each case has the same shape. Some parts of the program are **global**: the method, the scales, the control library, and the shared services everyone depends on. Other parts are **local**: the processes, sites, people, plans and incidents of one jurisdiction or unit. A few are **private**: personal data and restricted records that must never leave where they live.

OPTIONS

Three ways to run a multi-jurisdiction program

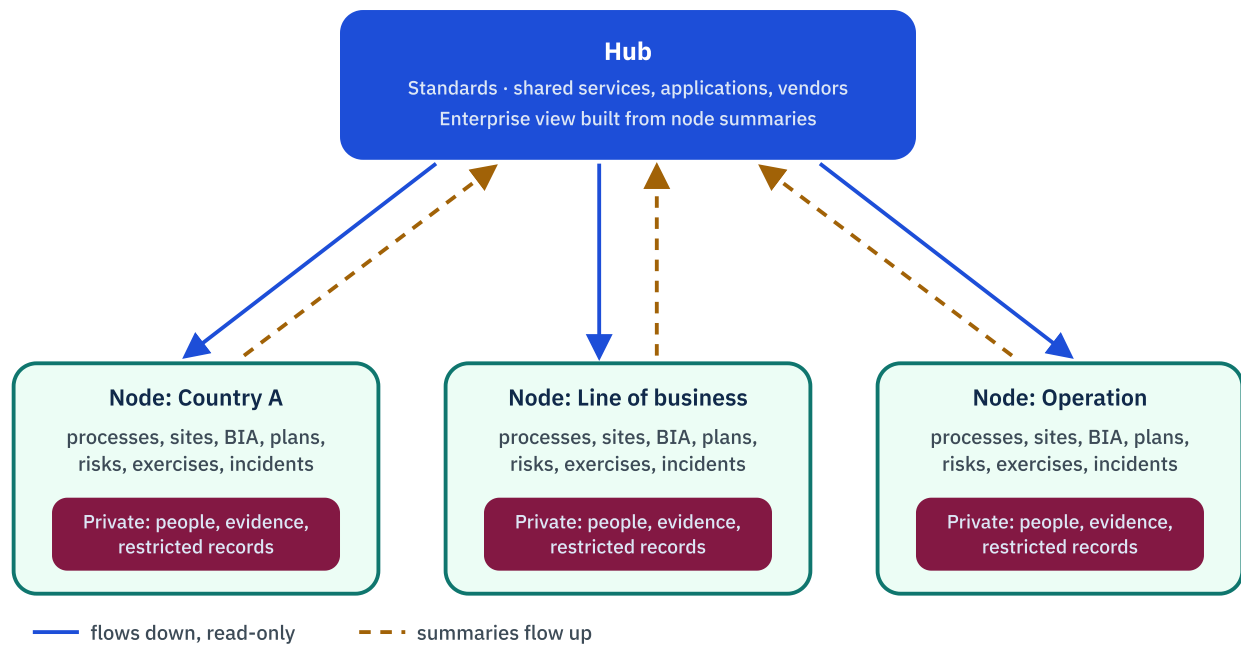
	One central database, partitioned	Federated platforms	Live query across databases
Data residency	Not met: all data sits in one place	Met: data stays in-country	Partly: stored locally, read live from outside
Air-gapped or sovereign sites	Only if the whole platform is	Designed for it: data moves as signed bundles	Needs live links between zones
Unit autonomy	Good, through access boundaries	Good	Good
Shared dependencies	Native	Through reference copies, with an "as of" date	Native while links are up, gone when they are down
Consistency	Strong	Strong: one owner per record, nothing to merge	Depends on every link
Best fit	One jurisdiction, several business units	Several jurisdictions or independent operations	Rarely; hard to secure and to certify

The practical answer combines the first two: **federation between jurisdictions, access boundaries inside each one**. A line of business that shares a jurisdiction with others can be a boundary inside one platform, and later become its own federated platform if a new law or an acquisition requires it.

THE MODEL

Global, local and private

In a federation, one platform acts as the **hub** and each jurisdiction, line of business or operation runs its own **node**: a complete resilience platform with its own database, hosted where its data must live. A single deployment is simply a federation of one.



Standards and shared entities are owned once at the hub. Each node owns its local records. Private data never leaves its node.

Where each kind of record lives

Class	Typical entities	How it moves
Global standard	Lists and categories, risk scales and appetite, impact categories, BIA time buckets, control library, plan templates, policies	Down to every node, read-only. A node may add local values but cannot change or remove global ones.
Global shared	Enterprise applications and IT services, shared services such as HR, payroll and facilities, global vendors and contracts, enterprise scenarios and risks, regulations	Down as read-only reference copies , only to the nodes that use them. Local records link to them like any other entity.
Local	Locations, departments, processes, local applications, BIA, plans, local risks, exercises, incidents	Stay in the node. An agreed summary goes up.
Private	Employees and contacts, home addresses and emergency contacts, evidence files, restricted risks	Never leave the node. The hub sees counts at most.

The common-functions case for public agencies is the "global shared" row. HR, IT, facilities and finance are built once at the hub. Every operation's plans and BIA depend on them through reference copies, and the hub can see, from the summaries, which operations depend on each common function and how tight their recovery objectives are.

Six rules that keep a federation simple

- 1 One owner per record.**

Every shared record has exactly one owning platform. Only the owner edits it; everyone else holds a read-only copy. Nothing ever needs to be merged, so two sites can never hold two conflicting versions of the truth. A node that needs a change raises a request, and the owner's approval team decides.
- 2 Data stays where it lives; summaries travel.**

What goes up is a defined summary per entity: for example, a process's criticality and recovery objectives, a risk's score and treatment status, a plan's last exercise result. Every field carries a data class, and each node's policy decides which classes may leave before anything is signed.
- 3 Standards travel first.**

An enterprise heat map or BIA roll-up only means something if every node uses the same scales and categories. Global standards are the first thing a federation shares, and the cheapest.
- 4 The customer holds the keys.**

Bundles between platforms are signed and encrypted with keys the organization owns. Any carrier can transport them, including removable media for an air-gapped site, but only the signature is trusted.
- 5 Honest about time.**

Every reference copy and every summary shows the date it is "as of". A federation never presents a copy as live data. Updates arrive in numbered order, and a missing or repeated update is refused and flagged.
- 6 Room to change shape.**

A business unit can start as an access boundary inside one platform and later become its own node, or move from one node to another, by transferring ownership of its records. Reorganizations, acquisitions and new residency laws should not mean a new implementation.

IN PRACTICE

Three scenarios

A multinational with in-country data rules

Head office runs the hub and sets the method. Each region with residency rules runs a node in-country. Employee data, sites and incidents stay there. Head office sees every critical service's recovery objectives, dependencies and exercise results across regions, without holding any personal data from them.

A group of autonomous businesses

Businesses that share a jurisdiction run as access boundaries inside one platform: each sees its own records plus the group's standards and shared services. A business in another jurisdiction, or one being prepared for sale, runs as its own node. The group's roll-up uses the same scales for all of them.

An agency with independent operations

The agency's common functions (HR, IT, facilities, finance) are built once at the hub. Each operation plans independently on its own node and links its processes to those common functions. When a common function fails, every dependent operation is identified at once.

When something goes wrong across borders

Incidents are the one flow that cannot wait for a scheduled update. A federation needs a small, fast incident status channel: which incident, how severe, which shared entities are affected, current status. Full detail stays local. An incident on a shared entity, such as an enterprise ERP or a global vendor, is raised by its owner and pushed to every node that depends on it, so each local team sees its own impact immediately. Where a site is fully air-gapped, the channel falls back to manual transfer, and the platform says so rather than assuming connectivity.

FOR BUYERS

Questions to ask any platform about federation

- ☐ **Where does each kind of record physically live?** Ask for the placement of people, sites, incidents, evidence and restricted records, not only "the data".
- ☐ **What exactly leaves a jurisdiction?** Ask to see the summary fields, their data classes, and who can switch a class off.
- ☐ **Who can edit a shared record, and what happens when two sites disagree?** "We merge" is a warning sign; "it has one owner" is the answer to look for.
- ☐ **Who holds the signing and encryption keys?** The organization should, not the vendor.
- ☐ **Does it work when a site is air-gapped?** Ask how updates and incident status move without a live connection.
- ☐ **Are the scales the same everywhere?** If each site can redefine "critical", the enterprise roll-up is not comparable.
- ☐ **What happens at the next reorganization?** Ask how a unit moves from one site or platform to another without re-entering its data.

eRMA and federation

eRMA, the resilience platform from eBRP Solutions, runs on the customer's own infrastructure, including air-gapped networks, and takes in data from outside as signed bundles rather than live connections.

Access is granted through teams, sensitive risks can be restricted to the people who should see them, and a complete export pack carries a whole program from one platform to another. These are the foundations the federated model in this paper builds on.

Organizations planning a multi-jurisdiction program are invited to discuss their requirements with eBRP Solutions.

eRMA by eBRP Solutions

Resilience since 2002.

Discuss your federated program: ebrp.net/demo