Rete, datacenter, energia e sovranità digitale

Gianluca Mazzini
Univesità di Ferrara

La rete ridondata non è necessariamente resiliente

La continuità dipende meno dal numero di collegamenti e più dall'indipendenza verificabile dei modi di guasto.

PUNTO DI TESI

Due circuiti acquistati da operatori diversi possono condividere lo stesso cavidotto, la stessa centrale, lo stesso PoP, lo stesso apparato di terminazione, la stessa alimentazione o la stessa squadra di manutenzione. In quel caso la ridondanza è commerciale, non sistemica.

VERIFICHE MINIME

- Disgiunzione fisica dei tracciati: tratte, scavi, ponti, attraversamenti, gallerie, cavidotti e camerette.
- Disgiunzione logica e ottica: anelli, lambde, ODF, patching, apparati attivi, sistemi di controllo e piani di routing.
- Disgiunzione dei punti di concentrazione: PoP, centrali, data center, carrier hotel, sedi di raccolta e nodi di interconnessione.
- Disgiunzione operativa: manutentori, procedure di escalation, finestre di lavoro, reperibilità, priorità in emergenza.
- Disgiunzione energetica: alimentazioni, cabine, UPS, generatori, autonomia locale e priorità di rialimentazione.

Messaggio: la resilienza di rete va auditata come catena di dipendenze, non dichiarata come somma di collegamenti.

Percorsi realmente disgiunti: cosa significa in pratica

La disgiunzione deve essere progettata, documentata, testata e mantenuta nel tempo.

CRITERI TECNICI

- Mappare ogni percorso fino al livello fisico: tratta, condotto, cameretta, edificio, sala, rack, ODF, apparato e alimentazione.
- Esplicitare i punti di possibile convergenza: accesso urbano, ingresso edificio, attraversamento obbligato, nodo metropolitano, centrale, dorsale.
- Evitare “diversità apparente”: fornitori diversi che subappaltano lo stesso tratto o usano la stessa infrastruttura passiva.
- Prevedere audit periodici: un percorso disgiunto oggi può cessare di esserlo dopo lavori, razionalizzazioni o migrazioni di rete.
- Introdurre clausole contrattuali di trasparenza: tracciati, cambiamenti, manutenzioni programmate, eventi incidentali e subfornitori.

INDICE MINIMALE DI SOVRAPPOSIZIONE

Un modo semplice per rendere misurabile il tema è rappresentare ogni percorso come insieme di componenti.

$O_{AB} = |A \cap B| / |A \cup B|$

$O_{AB} = 0$ indica percorsi completamente disgiunti. Valori maggiori di zero indicano condivisione di componenti e quindi rischio di guasto comune. La metrica non sostituisce l’analisi ingegneristica, ma obbliga a dichiarare cosa si considera davvero “separato”.

Rete propria, IRU e condipendenza

La scelta non è ideologica: dipende da criticità dei servizi, sovranità operativa richiesta e rischio di dipendenze comuni.

RETE PROPRIA: QUANDO PESA IL CONTROLLO

- Serve quando la priorità è governare manutenzione, tempi di intervento, evoluzione tecnologica, sicurezza fisica e priorità in emergenza.
- Consente maggiore visibilità su apparati, configurazioni, degrado, capacità residua, vulnerabilità operative e ciclo di vita degli asset.
- Ha senso per tratte critiche, accessi a siti essenziali, interconnessioni fra nodi strategici e segmenti in cui il lock-in operativo sarebbe rischioso.

IRU O RETE NOLEGGIATA: QUANDO PUÒ ESSERE RESILIENTE

- È utile se il tracciato è realmente alternativo e se il contratto dà visibilità su punti di concentrazione, subfornitori e manutenzioni.
- Richiede SLA verificabili, diritto di audit, notifica preventiva dei cambiamenti e obbligo di comunicare eventi che alterano la disgiunzione.
- Non va valutata solo per costo al chilometro: va pesata in base al rischio residuo che lascia sui servizi critici.

CONDIPENDENZA

La condipendenza è la situazione in cui reti formalmente diverse condividono fattori di rischio: energia, PoP, apparati, attraversamenti, manutentori, sistemi di gestione, vincoli territoriali o autorizzativi. La resilienza cresce quando si riducono queste dipendenze comuni, non quando si moltiplicano i contratti.

Backup radio e continuità: progettare il servizio minimo

Il backup non deve replicare tutto: deve sostenere ciò che permette all'organizzazione di restare viva.

PRINCIPIO DI PROGETTO

Il backup radio è efficace se viene dimensionato sul "minimum viable service": l'insieme minimo di applicazioni, flussi dati, canali di controllo e funzioni operative che devono rimanere attivi durante una crisi. Usarlo come copia povera della rete primaria porta quasi sempre a saturazione o falsa sicurezza.

VERIFICHE ESSENZIALI

- Capacità reale sotto stress: banda utile, latenza, jitter, perdita pacchetti, priorità del traffico e comportamento in congestione.
- Dipendenze del sito radio: alimentazione, batterie, visibilità ottica, autorizzazioni, condizioni meteo, frequenze, interferenze e sicurezza.
- Classificazione del traffico: controllo, autenticazione, gestione incidenti, comunicazioni operative, servizi al pubblico, sistemi interni.
- Piani di degradazione: cosa resta online, cosa passa in sola lettura, cosa viene rallentato, cosa viene sospeso.
- Test periodici realistici: failover non annunciati, esercizi sotto carico, ritorno alla normalità, verifica delle procedure di NOC/SOC.

Domanda guida: il backup sostiene i servizi davvero critici o sostiene solo l'idea astratta di avere una seconda via?

Energia come leva attiva: spegnere ciò che non serve

La novità non è solo “avere autonomia energetica”, ma usare l’energia come variabile di governo della resilienza.

CAMBIO DI PARADIGMA

Nella continuità tradizionale l’energia è trattata come requisito passivo: UPS, gruppi elettrogeni, batterie, contratti di fornitura. Nella resilienza sistemica diventa una leva attiva: decidere quali componenti alimentare, quali degradare e quali spegnere per preservare il nucleo critico più a lungo.

COSA SIGNIFICA OPERATIVAMENTE

- Misurare il consumo energetico per servizio, non solo per sito o apparato: capire quale funzione assorbe energia e quale valore produce.
- Definire livelli di servizio energetico: pieno regime, servizio essenziale, servizio minimo, conservazione, spegnimento controllato.
- Spegnere carichi non essenziali in modo orchestrato: elaborazioni batch, ambienti non produttivi, funzioni differibili, replica non urgente, servizi a bassa priorità.
- Proteggere l’autonomia dei nodi critici: non tutti i siti devono durare uguale; devono durare di più quelli da cui dipendono servizi ad alto valore.
- Integrare energia e rete: un link ridondato è utile solo se anche gli apparati, i PoP intermedi e il sito di destinazione restano alimentati.
- Testare lo spegnimento come procedura normale: non deve essere improvvisato durante l’emergenza, deve essere reversibile e tracciabile.

Messaggio: l’efficienza energetica non è solo sostenibilità; è capacità di sopravvivenza operativa.

Modello decisionale minimale: rischio, costo, energia

Una lettura pratica: proteggere ciò che sostiene servizi critici, alimentare solo ciò che serve, spegnere il resto in modo controllato.

Servizi

V_i = valore/criticità
L_i = perdita per ora di fermo
B_i min = banda minima
RTO_i max = ripristino massimo
E_i min = energia essenziale
→ **classifica cosa deve restare vivo**

Componenti

p_j = probabilità di guasto
$MTTR_j$ = tempo di ripristino
P_j = potenza assorbita
H_j = autonomia disponibile
C_j = costo di protezione
→ **misura cosa consuma e cosa rompe**

Dipendenze

a_{ij} = 1 se il servizio i
dipende dalla componente j
a_{ij} = 0 altrimenti
O_AB = sovrapposizione tra percorsi
→ **identifica guasti comuni e condipendenze**

Obiettivo
meno rischio residuo
energia e costi sostenibili

Calcolo della priorità

$K_j = \Sigma_i V_i \times a_{ij}$
Una componente ha priorità alta se sostiene molti servizi ad alto valore.
Con energia limitata, K_j diventa anche priorità di alimentazione.

Leve operative

Proteggere: ridondare, separare, presidiare, auditare.
Alimentare: garantire autonomia ai nodi essenziali.
Degradare/spegnere: carichi non essenziali, batch, ambienti non produttivi, repliche non urgenti.

Decisione: minimizzare rischio residuo + costo + energia consumata

$B_i \geq B_i$ min
i servizi essenziali hanno banda sufficiente

$RTO_i \leq RTO_i$ max
i servizi critici rientrano nei tempi

O_AB ≤ O max
i percorsi ridondanti non condividono troppo

$H_j \geq H_j$ min
i nodi essenziali hanno autonomia adeguata

Datacenter: la resilienza non coincide con avere piu macchine

Il punto critico e governare dipendenze, capacita essenziale, energia e tempi di ripristino lungo tutta la catena applicativa.

SCHEMA LOGICO DI RIFERIMENTO

Utenti / canali di accesso -> bilanciatore di ingresso -> servizi applicativi e produttori di dati -> code, cache e sistemi di integrazione -> database -> cloud e servizi avanzati -> osservabilita, sicurezza e orchestrazione.

ELEMENTI CRITICI DA ESPLICITARE

- Bilanciatore di ingresso: se cade o degrada, anche servizi sani diventano non raggiungibili; va pensato come piano di controllo del traffico, non come componente neutra.
- Servizi applicativi e produttori di dati: non tutti hanno la stessa criticita; alcuni devono restare sempre attivi, altri possono scalare a freddo o spegnersi.
- Database: spesso e il vero vincolo; consistenza, replica, latenza, lock, backup, restore e capacita di lavorare in sola lettura determinano la continuita reale.
- Cloud esterno: abilita funzioni avanzate, ma introduce dipendenza da API, identita, rete, quote, regioni, compliance, costi e tempi di ripristino non completamente governabili.
- Energia e raffreddamento: diventano variabili di governo del servizio; non servono solo a tenere acceso tutto, ma a scegliere cosa resta acceso.
- Osservabilita e test: senza telemetria e prove ricorrenti si confonde disponibilita dichiarata con resilienza dimostrata.

Tesi: il datacenter resiliente e un sistema capace di degradare senza collassare, non un insieme di componenti duplicate.

Always-on e on-demand: decidere cosa deve restare acceso

La domanda non e quante macchine tenere accese, ma quale livello minimo di servizio deve sopravvivere in ogni scenario.

CLASSIFICAZIONE OPERATIVA

SEMPRE ACCESI

Componenti senza cui il servizio essenziale non esiste: bilanciatori critici, identita minima, DNS interno, logging vitale, database primari o repliche calde, code essenziali, controllo rete e sicurezza.

ACCESI A CALDO

Capacita gia pronta ma riducibile: pool applicativi minimi, repliche pronte, cache essenziali, worker critici, nodi di integrazione con partner o cloud.

ACCESI ALLA BISOGNA

Capacita elastica o differibile: batch, analytics, training, ambienti non produttivi, servizi di supporto, funzioni avanzate non essenziali, risorse cloud scalabili.

CRITERI DI SCELTA

- Valore del servizio: cosa genera danno immediato se non disponibile e cosa puo attendere.
- Tempo di riaccensione: una macchina on-demand e utile solo se il tempo di bootstrap e compatibile con RTO e picco di domanda.
- Stato applicativo: i servizi stateless si scalano piu facilmente; quelli stateful richiedono replica, consistenza e procedure di ripartenza sicure.
- Energia consumata per valore prodotto: tenere acceso cio che non serve riduce autonomia, aumenta calore e sottrae margine ai servizi critici.
- Rischio operativo: spegnere deve essere una procedura progettata, automatizzata, reversibile e osservabile; non una decisione manuale presa in crisi.

Risultato atteso: un “profilo energetico di continuita” che associa a ogni scenario un set preciso di componenti accese, degradate o spente.

Risparmio energetico intelligente: efficienza come resilienza

La leva energetica diventa parte del controllo applicativo: consumare meno per durare di piu e proteggere il nucleo vitale.

COSA MISURARE

- Consumo per servizio e non solo per rack: legare potenza assorbita, traffico, transazioni, storage, CPU, GPU e memoria al valore operativo prodotto.
- Costo energetico della ridondanza: una replica sempre calda puo essere necessaria, ma va giustificata rispetto a rischio ridotto e autonomia persa.
- Energia nascosta: raffreddamento, apparati di rete, storage inattivo, ambienti di test, log non essenziali, pipeline dati e funzioni cloud sempre abilitate.
- Margine di autonomia: ore di funzionamento residue per cluster, sala, sito e servizio essenziale.

COME AGIRE

- Autoscaling consapevole della criticita: scalare non solo su carico, ma su priorita, scenario di crisi, costo energetico e disponibilita delle dipendenze.
- Spegnimento selettivo e programmato: ambienti non produttivi, job batch, funzioni differibili, repliche non critiche, cache secondarie, analytics e servizi ausiliari.
- Modalita degradata progettata: passaggio a sola lettura, riduzione frequenza aggiornamenti, compressione dei flussi, queueing controllato, disattivazione funzioni avanzate.
- Policy automatiche ma governate: soglie, approvazioni, rollback, tracciabilita, simulazioni e test sotto carico.
- Integrazione con facility: orchestrazione applicativa coerente con UPS, generatori, raffreddamento, PUE, batterie, microgrid e priorita di alimentazione.

Cosa ridondare, dove: non solo secondo datacenter

Avere piu siti non basta: contano distanza, rete, energia, dominio di guasto, latenza, dati e capacita di operare.

SCELTE DI RIDONDANZA

NELLO STESSO SITO

Ridondanza di apparati, alimentazioni, rack, storage, network fabric, bilanciatori e nodi applicativi. Protegge dal guasto locale, non dal sito indisponibile.

SECONDO SITO VICINO

Utile per bassa latenza, replica sincrona o quasi sincrona, continuita operativa rapida. Puo condividere rischi territoriali, energetici e di rete.

SITO REMOTO / CLOUD

Protegge da eventi territoriali piu ampi, ma introduce latenza, costi di replica, vincoli di sovranita, dipendenza da provider e complessita di failback.

CRITERI DI POSIZIONAMENTO

- Dominio di guasto: evitare che due datacenter condividano la stessa cabina elettrica, gli stessi carrier, la stessa area di rischio idrogeologico o lo stesso punto di concentrazione.
- Collegamenti: percorsi realmente disgiunti, capacita sufficiente in emergenza, piani di routing verificati, test di isolamento e procedure di riconvergenza.
- Dati: decidere cosa replicare in sincrono, cosa in asincrono, cosa ricostruire, cosa archiviare e cosa mantenere solo localmente per sicurezza o normativa.
- Operativita: personale, accessi, ricambi, runbook, credenziali, NOC/SOC, fornitori e possibilita di intervenire anche quando i sistemi centrali sono degradati.
- Energia: autonomia differenziata per sito; un sito secondario senza energia autonoma e solo una promessa di continuita.

Cloud come dipendenza: quanto e bloccante una componente esterna

Il cloud non e un blocco unico: bisogna distinguere funzioni critiche, funzioni avanzate e funzioni sostituibili.

ANALISI DI BLOCCANZA

BLOCCANTE

Identita, chiavi, API indispensabili, database gestiti, code centrali, storage primario o funzioni senza fallback. Se non disponibili, il servizio si ferma.

DEGRADANTE

AI, analytics, arricchimenti, notifiche, ricerca avanzata, dashboard, funzioni di supporto. Il servizio puo continuare con capacita ridotta o qualita inferiore.

SOSTITUIBILE

Funzioni con alternativa locale, provider secondario, cache, modalita offline, code differite o riconciliazione successiva.

DOMANDE DI CONTROLLO

- Cosa succede se la componente cloud non risponde per 15 minuti, 2 ore, 24 ore?
- Il servizio puo funzionare in modalita locale, in sola lettura, con cache, con code differite o con logica semplificata?
- Le credenziali, le chiavi, i token e i sistemi di identity sono ancora disponibili se il provider o la rete verso il provider sono degradati?
- Esistono limiti di quota, rate limit, costi di picco o vincoli contrattuali che diventano critici proprio durante la crisi?
- Il failover verso altro cloud o verso on-prem e testato anche nel ritorno alla normalita, cioe nel failback e nella riconciliazione dei dati?

Regola pratica: ogni dipendenza cloud deve avere una classe di bloccanza, una modalita degradata e un test periodico di indisponibilita simulata.

Modello minimale: ottimizzare resilienza, energia e continuità

Il datacenter resiliente non massimizza tutto: sceglie quali servizi tenere vivi, quali degradare e quali spegnere, rispettando vincoli di rischio, energia e dipendenze.

VARIABILI PER SERVIZIO

Per ogni servizio i

V_i = valore / criticità
L_i = perdita per ora di stop
B_i = capacità minima
E_i = energia minima per restare vivo

Stati possibili s

0 = spento
1 = degradato / sola lettura / coda differita
2 = pieno regime

$x_{i,s}$ indica lo stato scelto per il servizio i

Dipendenze

$a_{i,j} = 1$ se il servizio i dipende dalla componente j

Componenti j: bilanciatore, nodi app, database, code, cache, cloud, rete, energia, storage

RISCHIO E BLOCCANZA

Criticità della componente

$C_j = \Sigma_i \; V_i \cdot a_{i,j}$

Misura quanto è pericoloso perdere la componente j perché molti servizi critici dipendono da essa.

Bloccanza del cloud

$K_{i,cloud}$ = impatto se la funzione cloud manca

0 = sostituibile
1 = degradante
2 = bloccante

Rischio atteso del servizio

$R_i = P_{indisp,i} \cdot L_i \cdot T_i$

Dove $P_{indisp,i}$ cresce con guasti comuni, dipendenze non ridondate e componenti cloud bloccanti.

DECISIONE OTTIMALE

Funzione obiettivo

$\min \Sigma_i R_i + \lambda \Sigma_j \text{Costo}_j + \mu \Sigma_j \text{Energia}_j$

Ridurre rischio, costo e consumo insieme.

Vincoli minimi

Servizi vitali: $x_{i,s} \geq$ degradato
Capacità: $B_i \geq B_{i,min}$
RTO/RPO: entro soglie dichiarate
Energia: autonomia ≥ ore richieste
Cloud: nessuna funzione bloccante senza fallback

Output decisionale

- cosa resta sempre acceso
- cosa si accende a caldo
- cosa si spegne
- cosa si ridonda su altro sito
- cosa deve avere fallback locale

L'ottimo non è "più datacenter" o "più cloud": è la configurazione minima che mantiene vivo il nucleo operativo, consuma meno energia possibile e rende misurabile il rischio residuo.

Mettere insieme rete e datacenter: una sola catena di continuita

La continuita non nasce dalla somma di due ridondanze separate, ma dal coordinamento tra traffico, calcolo, dati, energia e procedure.

SEQUENZA DA GOVERNARE

Utente / sede / servizio esterno -> rete di accesso -> percorsi geografici -> punto di ingresso -> bilanciamento -> applicazioni -> dati -> cloud esterno -> osservabilita e controllo -> energia.

PUNTI CRITICI DI ACCOPPIAMENTO

- Il datacenter puo essere sano ma non raggiungibile: rete, DNS, identity, routing e bilanciamento diventano parte del servizio applicativo.
- La rete puo essere ridondata ma inutilizzabile se tutti i percorsi portano allo stesso punto energetico, allo stesso cloud, allo stesso database o allo stesso dominio di autenticazione.
- Il failover applicativo richiede banda, latenza e stato dei dati coerenti: non basta avere un secondo sito se non e collegato nel modo giusto.
- La degradazione deve essere coordinata: ridurre traffico, spegnere servizi non essenziali, abbassare qualita, sospendere batch e proteggere database e code.
- La telemetria deve restare disponibile anche in crisi: senza misura non si capisce se il problema e rete, calcolo, dati, energia, cloud o configurazione.

Tesi: la resilienza va progettata come proprieta end-to-end, non come requisito separato di rete o di datacenter.

Architettura integrata: domini di guasto e domini di decisione

Il progetto deve separare cio che puo rompersi insieme e unire cio che deve decidere insieme.

DOMINIO RETE

Percorsi fisici, carrier, PoP, routing, DNS, bilanciatori geografici, capacita minima, priorita del traffico, isolamento e riconvergenza.

DOMINIO DATACENTER

Siti, cluster, storage, database, code, cache, identity, osservabilita, orchestrazione, capacita calda e capacita accendibile.

DOMINIO ENERGIA

UPS, gruppi, batterie, raffreddamento, carico essenziale, spegnimento controllato, autonomia per servizio, priorita di alimentazione.

REGOLE DI PROGETTO

- Separare i domini di guasto: due siti non sono indipendenti se condividono rete, energia, cloud, personale operativo, credenziali o supplier critici.
- Unire i domini di decisione: rete, calcolo, dati ed energia devono ricevere lo stesso segnale di stato e applicare la stessa politica di continuita.
- Definire profili operativi: normale, degrado controllato, emergenza energetica, isolamento cloud, isolamento rete, ripristino e failback.
- Associare a ogni profilo una configurazione: link prioritari, servizi attivi, repliche abilitate, cloud usabile, traffico ammesso e funzioni spente.
- Misurare il rischio residuo su catene di servizio, non su componenti isolate.

Principio operativo: cio che si rompe insieme non deve essere considerato ridondanza; cio che decide insieme deve essere orchestrato esplicitamente.

Continuita end-to-end: dal failover tecnico alla configurazione di crisi

La domanda non e solo dove si sposta il carico, ma quale servizio minimo rimane vivo e con quali dipendenze.

INGRESSO

Bilanciamento locale e geografico, DNS resilienti, percorsi disgiunti, protezione DDoS, instradamento del traffico essenziale.

ESECUZIONE

Pool applicativi minimi, scalabilita controllata, spegnimento batch, modalita sola lettura, code differite, priorita tra servizi.

DATI

Replica sincrona/asincrona, RPO esplicito, backup verificati, riconciliazione, consistenza, fallback locale e ricostruzione.

ESTERNI

Cloud, identity, API, chiavi, quote, regioni, fornitori, compliance, alternativa locale o degrado funzionale.

DALLA PROCEDURA DR AL PIANO DI CONTINUITA

- DR classico: ripristinare un sistema dopo un evento, spesso con obiettivo tecnico e verifica episodica.
- Business continuity reale: mantenere in vita il nucleo operativo durante l evento, anche degradando qualita, capacita e funzionalita.
- Configurazione di crisi: insieme predefinito di servizi accesi, servizi degradati, servizi spenti, link prioritari, dati disponibili e cloud bypassabile.
- Failback: il ritorno alla normalita e parte del disegno; senza riconciliazione dei dati e controllo delle code si sposta il problema in avanti.
- Prove integrate: testare insieme rete, datacenter, energia, cloud, procedure e persone; non solo il singolo componente.

La resilienza matura non promette che nulla si fermi: promette che si ferma prima cio che vale meno e resta attivo cio che sostiene il servizio essenziale.

Sovranita digitale: ridurre dipendenze non governabili

Essere piu sovrani non significa fare tutto in casa, ma sapere cosa non puo dipendere da scelte terze non controllabili.

DIPENDENZE DA RENDERE VISIBILI

TECNOLOGICHE

Provider cloud, software proprietario, firmware, apparati, sistemi di orchestrazione, database gestiti, librerie, API, formati dati.

CONTRATTUALI

SLA non verificabili, lock-in, diritto di audit assente, costi di uscita, dipendenza da roadmap del fornitore, rinnovi e variazioni unilaterali.

GEOPOLITICHE

Paesi terzi, giurisdizioni, trasferimento dati, restrizioni export, sanzioni, controllo delle catene di fornitura, localizzazione e accesso.

COSA MODIFICARE NEI MODELLI

- Aggiungere una classe di sovranita a ogni componente: controllata direttamente, controllata contrattualmente, sostituibile, non sostituibile, esterna critica.
- Separare servizio e fornitore: progettare interfacce, dati e procedure in modo da poter cambiare provider o attivare un fallback senza riprogettare tutto.
- Mantenere un nucleo essenziale locale o sovrano: identity minima, chiavi, logging vitale, configurazioni, dati critici, procedure di accesso e controllo.
- Richiedere portabilita: export dei dati, standard aperti, infrastruttura come codice, automazioni trasferibili, backup indipendenti dal provider.
- Valutare la ridondanza giuridica oltre a quella tecnica: due cloud nello stesso paese terzo o sotto vincoli analoghi possono non essere realmente indipendenti.

Architettura piu sovrana: dove mettere controllo, alternative e fallback

La sovranita si costruisce scegliendo quali leve devono restare azionabili anche quando un fornitore, un paese o una piattaforma non lo sono.

CONTROLLO DIRETTO

Rete critica, configurazioni, chiavi, identity minima, repository, backup, runbook, monitoraggio essenziale, dati primari o copie verificabili.

CONTROLLO CONTRATTUALE

IRU, colocation, cloud, SaaS, manutenzione, licenze, supporto, diritti di audit, obblighi di trasparenza, localizzazione e uscita.

CONTROLLO PER SOSTITUIBILITA

Multi-provider, standard aperti, abstraction layer, replica indipendente, procedure di migrazione, formati portabili, fallback funzionale.

LINEE DI AZIONE

- Disegnare un perimetro minimo sovrano: il set di capacita senza cui non si puo governare la crisi anche se i servizi avanzati sono fuori uso.
- Ridurre funzioni bloccanti esterne: trasformare dove possibile una dipendenza bloccante in dipendenza degradante o sostituibile.
- Usare cloud e fornitori come capacita, non come unico piano di controllo: evitare che gestione, identita, chiavi e osservabilita siano tutti nello stesso dominio esterno.
- Tenere esercitata l opzione di uscita: backup ripristinabili altrove, configurazioni ricreabili, dati leggibili, contratti con exit plan e test periodici.
- Bilanciare efficienza e sovranita: centralizzare tutto riduce costi nel breve periodo, ma aumenta lock-in e impatto sistemico del guasto o della decisione terza.

Criterio guida: non tutto deve essere interno; tutto cio che e vitale deve essere governabile, verificabile o sostituibile.

Modello finale: ottimizzare resilienza, energia e sovranita

Il modello integra rete e datacenter aggiungendo un coefficiente di dipendenza esterna e un vincolo di governabilita.

Criticita della catena di servizio

Per ogni servizio i:
Valore_i, PerditaOra_i, RTO_i, RPO_i, BandaMin_i, EnergiaMin_i, QualitaMin_i.

La priorita non e la macchina: e la catena necessaria a mantenere vivo il servizio.

Dipendenza comune e sovrapposizione

Per ogni componente j:
Rete, sito, energia, cloud, database, identity, fornitore, paese, team operativo.

Sovrapposizione(A,B) = componenti comuni / componenti totali.

Sovranita della componente

Sovranita_j cresce se la componente e controllata, auditabile, sostituibile e localizzata in un dominio governabile.

DipendenzaEsterna_j cresce se e opaca, bloccante, non portabile o soggetta a decisioni terze.

Rischio atteso del servizio

Rischio_i = ProbStop_i x PerditaOra_i x Durata_i.

ProbStop_i aumenta quando la catena dipende da componenti comuni, non sostituibili, non governabili o energeticamente fragili.

Vincoli minimi

Banda >= BandaMin
Energia disponibile >= EnergiaMin
RTO <= RTO massimo
RPO <= RPO massimo
Sovrapposizione <= soglia
Dipendenza esterna bloccante <= soglia

Funzione obiettivo

Minimizzare:
Rischio residuo + costo + consumo energetico + lock-in

Garantendo:
servizio minimo, autonomia, portabilita, disgiunzione reale e capacita di spegnere cio che non serve.

LETTURA MANAGERIALE DEL MODELLO

Si investe prima sulle componenti che sostengono molti servizi critici, che consumano energia essenziale, che non hanno alternativa e che dipendono da domini esterni non governabili. La scelta ottima non massimizza la ridondanza: minimizza il rischio sistemico mantenendo sostenibilita energetica e sovranita operativa.