Scalable and Reliable Systems nell'era dell'Artificial Intelligence

Architetture, operations e resilienza

Vincenzo Calabrò · Luglio 2026

Obiettivi didattici

Al termine del modulo lo studente sarà in grado di:

1 · Metriche e lessico

Padroneggiare la tassonomia della dependability, le metriche SLI/SLO, la tail latency e la loro rilettura per i workload di AI.

2 · Sistemi per l'AI

Comprendere parallelismo 3D, fault tolerance del training su larga scala e architetture di inference serving per LLM.

3 · AI per i sistemi

Valutare criticamente AIOps e agenti autonomi: che cosa funziona in produzione, che cosa i benchmark smentiscono.

4 · Resilienza

Integrare threat model, forensic readiness e vincoli normativi (AI Act) nel progetto di sistemi affidabili e sicuri.

Percorso

Tre blocchi, un filo conduttore: la resilienza

Parte I — Sistemi scalabili e affidabili per l'AI

Training distribuito e parallelismo 3D · fault tolerance: checkpointing, detection, isolamento
Inference serving: batching, memoria, disaggregazione · MLOps come reliability engineering

Parte II — L'AI per la scalabilità e l'affidabilità dei sistemi

AIOps: dalla anomaly detection statistica ai LLM · incident management in produzione (RCACopilot)
Agenti autonomi e benchmark: AIOpsLab, ITBench, STRATUS

Reliability × Security — la prospettiva della resilienza

Threat model NIST, poisoning e prompt injection · un unico spazio dei guasti
Osservabilità, audit trail, forensic readiness e AI Act

Il caso di apertura: il pre-training di Llama 3 405B

54 giorni · cluster di 16.384 GPU NVIDIA H100 (Llama Team, AI @ Meta, 2024)

466
interruzioni del job registrate durante l'addestramento
di cui 47 pianificate

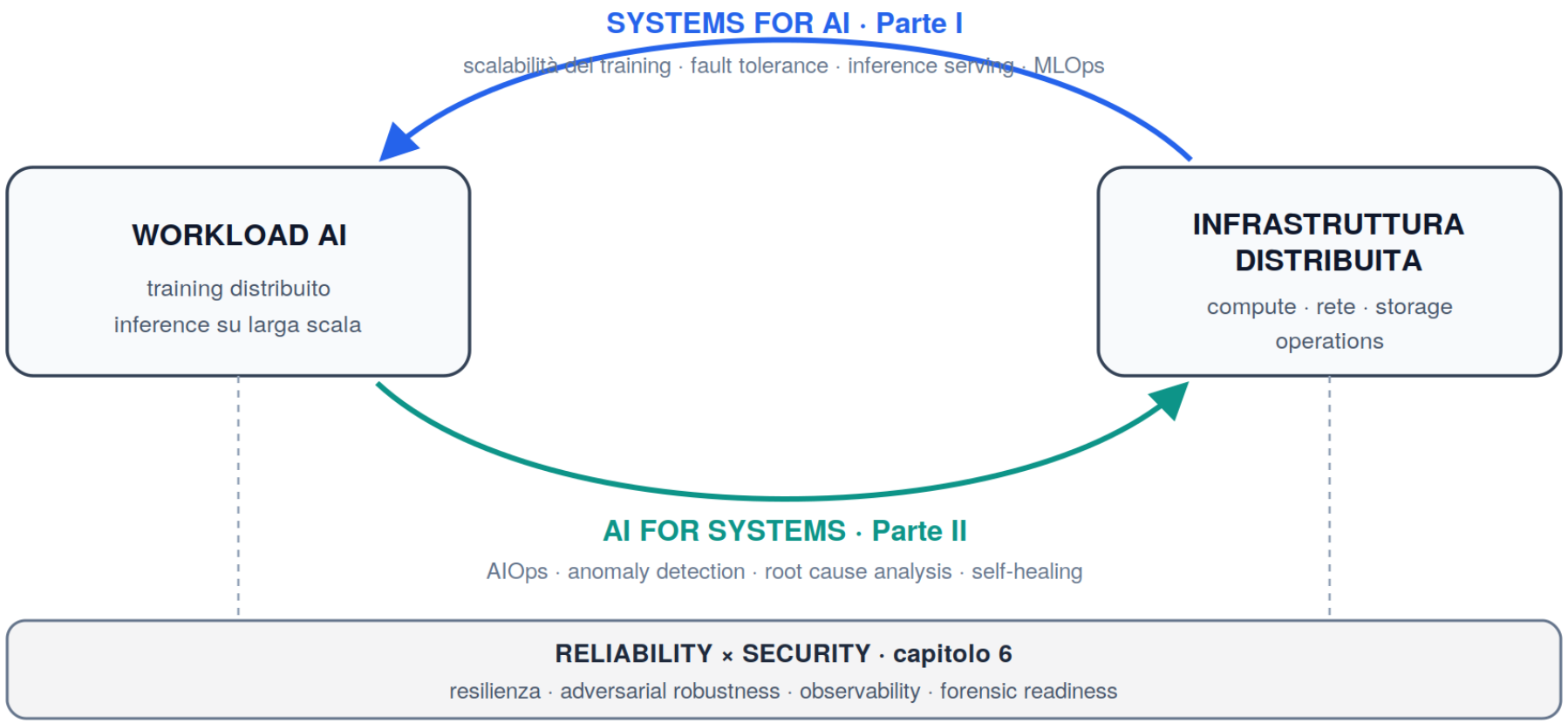
419
interruzioni impreviste: in media una ogni tre ore
Tabella 5 del paper

78%
degli eventi imprevisti riconducibile a guasti hardware (58,7% GPU)
confermati o sospetti

>90%
effective training time mantenuto nonostante tutto
tre soli interventi manuali

Alla scala attuale il guasto non è un'eccezione da prevenire: è una condizione operativa permanente da governare.

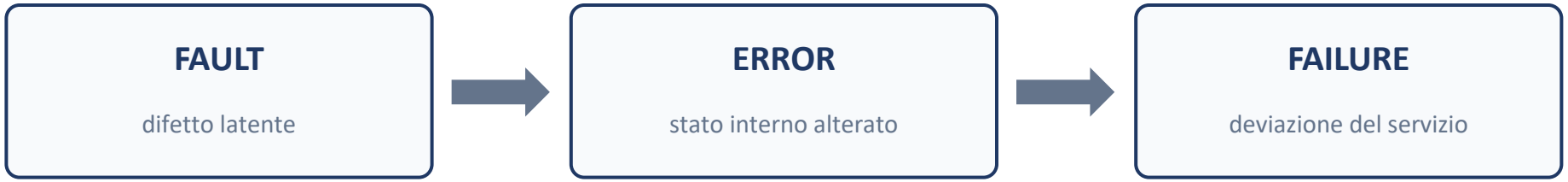
La doppia convergenza tra AI e systems engineering



Le due direttrici del corso e il piano trasversale della resilienza

Fondamenti: la tassonomia della dependability

Avizienis, Laprie, Randell, Landwehr — IEEE TDSC, 2004



- Attributi: availability, reliability, safety, integrity, maintainability

la security condivide availability e integrity e aggiunge la confidentiality

- Quattro mezzi: prevenzione, tolleranza, rimozione, previsione dei guasti
- I fault si classificano anche per intento: accidentali o dolosi

la saldatura reliability–security è già nel quadro fondativo — la riprenderemo

Quantificare l’affidabilità

Le grandezze di base e una conseguenza spesso trascurata

$$A = MTTF / (MTTF + MTTR)$$

- MTTF — tempo medio di funzionamento corretto
- MTTR — tempo medio di ripristino
- MTBF = MTTF + MTTR
- **Dimezzare l’MTTR vale quanto raddoppiare l’MTTF**

la pratica moderna investe sulla velocità di recovery, non solo sulla prevenzione

I «nove» di availability

99,9% → ~8 h 45 min di indisponibilità / anno
99,99% → ~52 min / anno
99,999% → ~5 min / anno

Le soglie si negoziano: ogni nove in più
ha un costo architetturale crescente.

SLI, SLO, SLA e l’error budget

L’operazionalizzazione del Site Reliability Engineering (Beyer et al., 2016)

SLI

Indicatore misurato.
Es.: frazione di richieste servite entro 300 ms; tasso di errori.

SLO

Obiettivo interno sull’indicatore.
Es.: il 99,9% delle richieste entro soglia nel trimestre.

SLA

Impegno contrattuale verso il cliente, con penali. Sempre meno stringente dello SLO.

Error budget = la quota di inaffidabilità ancora «spendibile» nel periodo (distanza SLI–SLO). Trasforma l’affidabilità da vincolo assoluto a risorsa finita, allocabile per bilanciare velocità di rilascio e stabilità.

La tirannia della coda: tail latency

Dean & Barroso, «The Tail at Scale» — CACM, 2013

63%

delle richieste subisce almeno una risposta lenta se un servizio interroga 100 backend e ciascuno è lento nell'1% dei casi: $1 - 0{,}99^{100} \approx 0{,}63$

- La media è quasi priva di significato: contano p99 e p99.9
- Il fan-out amplifica la coda per pura combinatoria
- Risposta «tail-tolerant»: hedged request, repliche selettive

in analogia con la fault tolerance: costruire un insieme prevedibile da parti che non lo sono

- Nei LLM la coda torna centrale: batching e lunghezza variabile dell'output

lo vedremo con TTFT e TPOT nel blocco serving

Scalability: i limiti teorici e la lettura moderna

Amdahl (1967) — strong scaling

Problema fisso, risorse crescenti.
La frazione seriale pone un tetto
invalicabile allo speedup:
con il 5% seriale, mai oltre 20×,
con qualunque numero di processori.

Gustafson (1988) — weak scaling

Risorse crescenti, problema crescente.
Nella pratica si affrontano istanze
più grandi: lo speedup va misurato
a carico scalato. È il regime del
training dei foundation model.

Lettura ingegneristica odierna: scalare = aumentare il throughput in proporzione alle risorse mantenendo pressoché costante il costo marginale per unità di lavoro.

Perché i workload AI cambiano le regole

Tre discontinuità strutturali rispetto ai sistemi web-scale

1 · Failure domain

Nel training sincrono l'intero cluster è un unico dominio di guasto: ogni step attende tutti gli acceleratori.

Il guasto non si isola:
si propaga per costruzione.

2 · Statistica

A parità di affidabilità del singolo componente, la frequenza dei guasti cresce ~linearmente con N.

Job da settimane su decine di migliaia di acceleratori.

3 · Economia

Ogni ora di stallo immobilizza hardware, energia e tempo di calendario senza precedenti.

Il costo del downtime passa dal mancato servizio al mancato progresso.

Conseguenza: availability → effective training time / ETTR; throughput → Model FLOPs Utilization (MFU); MTTR → detection + diagnosi + rescheduling + restore.

PARTE I

Sistemi scalabili e affidabili per l’AI

Training distribuito · fault tolerance · inference serving · MLOps

Training distribuito: i tre assi del parallelismo

Data parallelism

Modello replicato, batch ripartito. All-reduce dei gradienti a ogni step: è la sincronizzazione globale del sistema.

ZeRO partiziona stati, gradienti e parametri (Rajbhandari 2020).

Tensor parallelism

Le singole matrici di un layer divise tra acceleratori. Comunicazione intensissima: confinato entro il nodo, su interconnessioni NVLink.

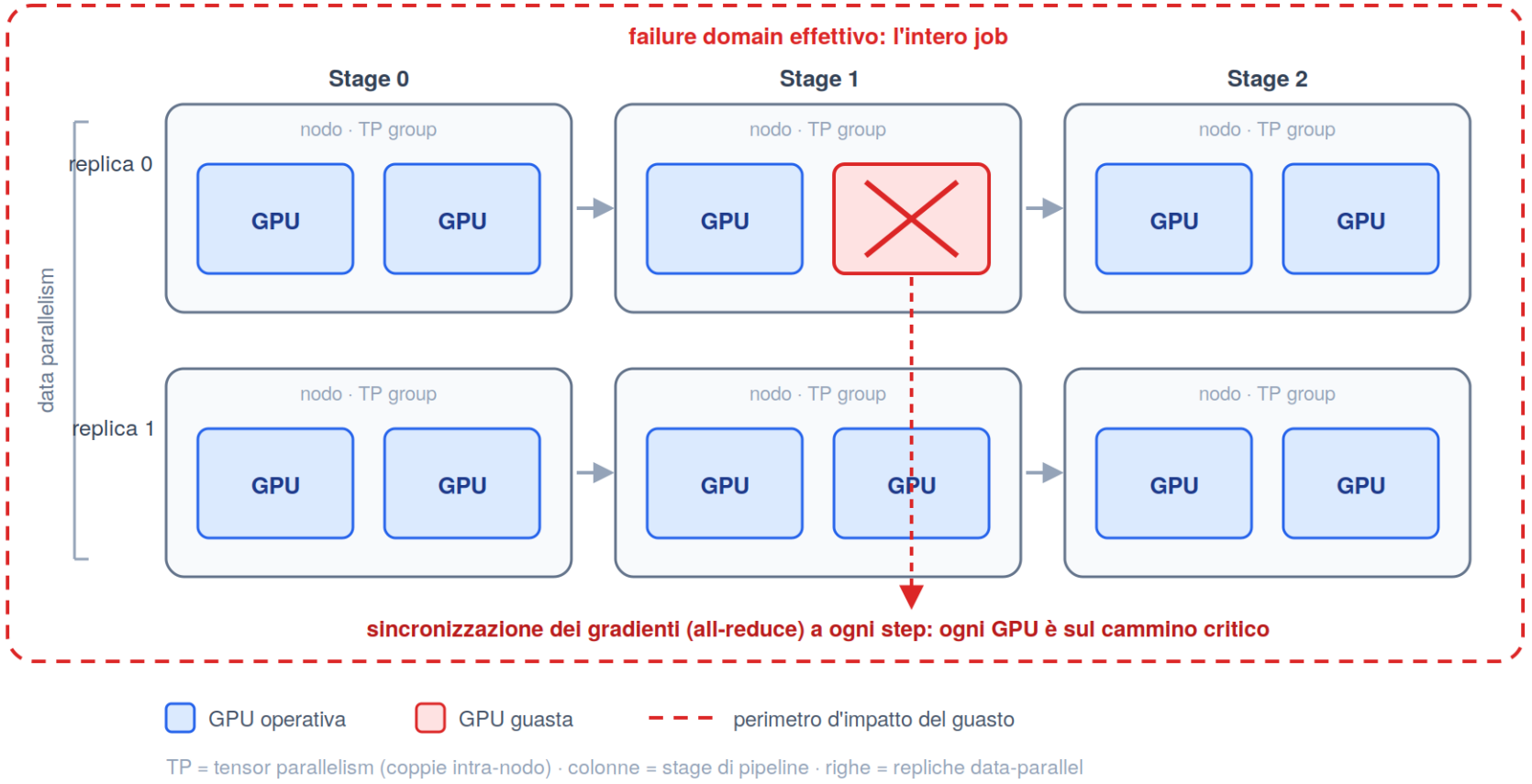
Megatron-LM (Narayanan 2021).

Pipeline parallelism

Layer in stadi consecutivi su gruppi diversi; micro-batch in sequenza per ridurre la «bolla» di inattività alle estremità di ogni step.

Composizione 3D con mappatura canonica: TP intra-nodo, PP tra nodi vicini, DP alla scala ampia. MegaScale: MFU 55,2% su 12.288 GPU, +34% vs baseline (Jiang et al., NSDI 2024).

L’inversione del failure domain



La griglia di parallelismo è anche la superficie di guasto: una GPU ferma blocca la barriera globale

Fault tolerance I — preservare lo stato: il checkpointing

- Ottimizzazione a tre variabili

frequenza dei salvataggi · throughput sottratto al training · lavoro perduto e da ricalcolare

- Stati da centinaia di GB ad alcuni TB: lo storage remoto è collo di bottiglia

frequenze basse → finestre di lavoro perduto ampie

- Svolta: primo livello di persistenza nella memoria CPU degli host

Gemini — SOSP 2023

Checkpoint in-memory con piazzamento quasi ottimo delle copie e traffico interfogliato con l'addestramento.

→ salvataggio a ogni iterazione, senza penalità di throughput
→ recovery oltre 13× più rapido (Wang et al., 2023)

ByteCheckpoint (Wan et al., 2024): checkpointing unificato lungo tutto il ciclo di sviluppo, anche al variare della configurazione di parallelismo.

Fault tolerance II — detection e isolamento

- I guasti più costosi non si annunciano: hang silenziosi, NaN, degradi

segnale di default: timeout delle collettive ~10 minuti — a 16.000 GPU è calcolo perduto su scala industriale

- Straggler: lento ≠ morto

sfugge ai rilevatori binari e impone il proprio passo al cluster; servono baseline per componente

- Llama 3: automazione quasi totale del recovery

solo 3 interruzioni su 419 con intervento manuale significativo

ByteRobust — SOSP 2025

Filosofia dichiarata: isolare in fretta conta più che localizzare con precisione.

Piattaforma di produzione con oltre 200.000 GPU · hot update aggregati, macchine warm di riserva, checkpointing fault-aware.

→ ETTR 97% su un job di 3 mesi a 9.600 GPU (Wan et al., 2025)

Inference serving: un regime operativo diverso

Dal calcolo unico e sincrono al flusso di richieste sotto SLO di latenza

Prefill — compute-bound

Elabora il prompt in parallelo,
satura il calcolo.

Metrica: TTFT — Time To First Token
(la reattività percepita).

Decode — memory-bound

Genera un token alla volta, limitato
dalla banda di memoria; il KV cache
cresce con la sequenza.

Metrica: TPOT — Time Per Output Token
(la fluidità della generazione).

Goodput = richieste al secondo servite nel rispetto di entrambi gli SLO (TTFT e TPOT). È la metrica di servizio di riferimento (Zhong et al., OSDI 2024). Complicazione strutturale: il costo di una richiesta dipende dalla lunghezza dell’output, ignota a priori.

L’evoluzione dei sistemi di serving, 2022–2024

Sistema	Sede	Idea chiave	Contributo
Orca (Yu et al.)	OSDI 2022	Continuous batching a livello di iterazione	Elimina le attese strutturali del batch a richiesta
vLLM (Kwon et al.)	SOSP 2023	PagedAttention: KV cache paginata e condivisibile	Frammentazione ~azzerata; throughput 2–4×
Sarathi-Serve (Agrawal et al.)	OSDI 2024	Chunked prefill, batching senza stalli	Attenua il trade-off throughput / tail latency
Splitwise (Patel et al.)	ISCA 2024	Fasi prefill e decode su pool di macchine distinti	Dimensionamento (ed equipaggiamento) per fase
DistServe (Zhong et al.)	OSDI 2024	Disaggregazione orientata al goodput	Massimizza le richieste entro SLO TTFT/TPOT

MLOps come reliability engineering

Kreuzberger, Kühl, Hirschl — IEEE Access, 2023 · Sculley et al. — NeurIPS, 2015

Concetto MLOps	Lettura reliability
Monitoraggio del data/model drift	SLI del modello: il degrado silenzioso diventa misurabile
Retraining automatico (continuous training)	Remediation: riporta la qualità entro l'obiettivo
Model registry con versioni e rollback	Punto di ripristino noto: il «checkpoint» del comportamento
Versionamento congiunto di codice, dati, modelli	Riproducibilità e attribuzione dei guasti

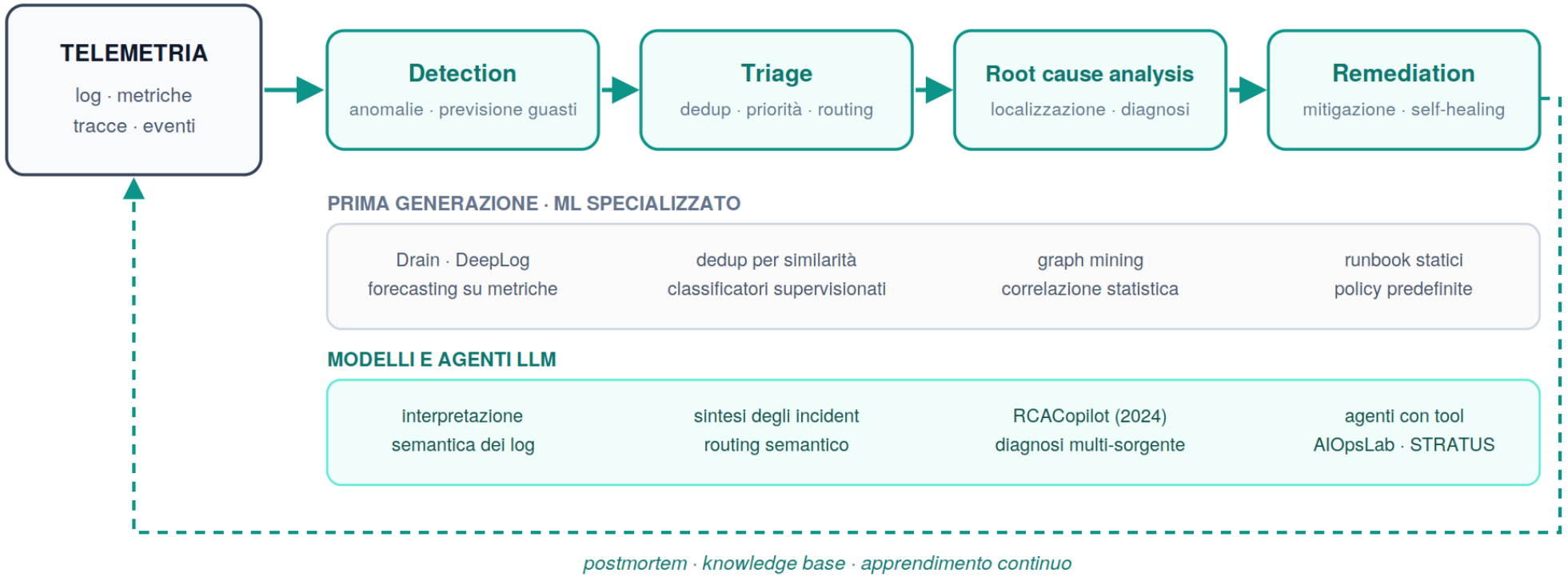
Il debito tecnico dei sistemi ML vive nei dati e nei confini tra componenti, non nel codice: «changing anything changes everything» (Sculley et al., 2015). Un modello accurato al rilascio degrada in silenzio quando la distribuzione degli input deriva.

PARTE II

L’AI per la scalabilità e l’affidabilità dei sistemi

AIOps · incident management con LLM · agenti autonomi e benchmark

La pipeline del failure management



Telemetria → detection → triage → root cause analysis → remediation, con il circuito di retroazione dei postmortem

Dalla prima generazione ai Large Language Model

Zhang et al., ACM Computing Surveys 2025: 183 lavori analizzati, 2020–2024

ML specializzato — i limiti

Un modello per compito, dati etichettati costosi
Fragile all’evoluzione dei formati di log
Ogni segnale nel proprio silo: niente correlazione log–metriche–tracce
Proliferazione di rilevatori → alert fatigue

Riferimenti: Drain (He 2017), DeepLog (Du 2017)

LLM — che cosa cambia

Semantica del testo semi-strutturato senza training dedicato
Correlazione multi-sorgente in un unico spazio
Adattamento few-shot a servizi nuovi
Interfaccia in linguaggio naturale

Cautele: risposte plausibili ma infondate → grounding su dati verificabili; costi e latenza

Caso di produzione: RCACopilot a Microsoft

Chen et al. — EuroSys 2024

1 · Routing

incidente → handler per tipo di alert

2 · Evidenze

raccolta diagnostica multi-sorgente

3 · Diagnosi

categoria di causa + narrazione

0,766

accuratezza di root cause analysis
su un anno di incidenti reali
EuroSys 2024

4+ anni

la componente di raccolta diagnostica
in produzione a Microsoft
EuroSys 2024

La lezione architetturale: al modello non si affida un compito aperto. L'affidabilità dell'assistente dipende dal grado in cui il flusso di lavoro lo vincola a evidenze raccolte e verificabili: il grounding come proprietà strutturale della pipeline.

Quanto sono autonomi davvero? I benchmark

AIOpsLab (MLSys 2025) · ITBench (Jha et al., ICML 2025) — 102 scenari in ambienti Kubernetes reali

11,4%
scenari SRE risolti dagli agenti allo stato dell'arte
ITBench, ICML 2025

25,2%
scenari CISO (compliance e sicurezza)
ITBench, ICML 2025

25,8%
scenari FinOps (ottimizzazione dei costi)
ITBench, ICML 2025

- AIOpsLab: valutazione end-to-end con fault injection e carico realistico

ambienti a microservizi riproducibili; compiti di detection, localizzazione, diagnosi, mitigazione

- **La distanza tra assistenza conversazionale e azione autonoma è ampia e misurabile**
- Chiudere il ciclo trasforma un problema di qualità delle risposte in un problema di sicurezza delle azioni

un'azione sbagliata va annullata — ammesso che sia annullabile

STRATUS e la Transactional No-Regression

Chen et al. — NeurIPS 2025

- Sistema multi-agente per la site reliability autonoma

agenti specializzati (detection, diagnosi, mitigazione) organizzati in una macchina a stati

- Specifica di sicurezza formale: Transactional No-Regression (TNR)

l'esplorazione delle azioni correttive è ammessa solo in forma transazionale, con la garanzia di non peggiorare mai lo stato del sistema

- La macchina a stati è anche un registro delle decisioni: auditabilità by design

≥1,5×

tasso di successo nella mitigazione vs agenti allo stato dell'arte

su AIOpsLab e ITBench

La disciplina di sicurezza non penalizza l'efficacia: la abilita. È la traduzione agentica di un principio antico — la reversibilità delle azioni come precondizione dell'automazione.

RELIABILITY × SECURITY

La prospettiva della resilienza

Threat model NIST · un unico spazio dei guasti · forensic readiness e AI Act

Il threat model: la tassonomia NIST AI 100-2 E2025

Vassilev et al., marzo 2025 — il riferimento per l'adversarial machine learning

- Cinque dimensioni di classificazione

tipo di sistema (PredAI / GenAI) · fase del lifecycle · obiettivi violati · capacità · conoscenza dell'attaccante

- Obiettivi: availability, integrity, privacy — più il misuse per la GenAI
- Novità 2025: supply chain, prompt injection indiretta, rischi agentici

Classe di attacco	Proprietà violata
Data poisoning (incl. backdoor)	Integrity (availability se degradante)
Evasion — adversarial examples	Integrity
Attacchi alla privacy	Confidentiality / privacy
Prompt injection (diretta/indiretta)	Integrity, misuse
Supply chain del modello	Integrity

Due attacchi da conoscere

Poisoning web-scale: praticabile

Carlini et al. — IEEE S&P 2024:

Split-view: domini scaduti che ospitano frammenti dei dataset distribuiti → ciò che il curatore validò ≠ ciò che i client scaricano

Frontrunning: modifiche calibrate sugli snapshot periodici (es. Wikipedia)

Dieci dataset di uso comune avvelenabili

Indirect prompt injection

Greshake et al. — AISec 2023:

Istruzioni ostili collocate nei contenuti che il modello recupera ed elabora (pagine web, documenti, email).

Ogni sorgente di contesto diventa un potenziale canale di comando.

Corollario per l'AIOps: un agente privilegiato legge log, ticket e alert influenzabili da terzi. Una riga di log confezionata ad arte è l'equivalente funzionale di un input di comando non fidato.

Un unico spazio dei guasti

proprietà × origine	**FAULT ACCIDENTALE** *dominio della reliability*	**FAULT INTENZIONALE** *dominio della security*
AVAILABILITY	guasti hardware e di rete straggler · esaurimento risorse	denial of service availability poisoning input a costo massimizzato
INTEGRITY	silent data corruption bit flip · bug software	data e model poisoning evasion (adversarial examples) prompt injection
CONFIDENTIALITY	leakage da misconfigurazione memorizzazione involontaria	membership inference model extraction ricostruzione dei dati

stessa disciplina di risposta: rilevazione → isolamento → ripristino → evidenza

ciò che cambia è il modello del fault: casuale e indipendente vs adattivo e worst-case

Stessa disciplina di risposta; cambia il modello del fault: casuale e indipendente vs adattivo e worst-case

L’asimmetria che resta

Strumenti condivisi, modelli di minaccia distinti

Fault accidentale

Casuale e indipendente.

Contro di esso la ridondanza statistica
(repliche, maggioranze, checksum)
è efficace e quantificabile.

Verifica: chaos engineering.

Fault avversario

Adattivo, deliberatamente worst-case:
conosce le difese, ne cerca i bordi.

La robustezza al rumore non dice nulla
sulla robustezza a perturbazioni ottimizzate.

Verifica: red teaming.

Due discipline della stessa fault injection: guasti iniettati secondo il modello accidentale o secondo quello avversario. I limiti teorici (es. tolleranza bizantina) valgono solo entro le ipotesi per cui sono derivati.

Forensic readiness dei sistemi AI

Rowlingson, 2004: predisporre la raccolta di evidenza utilizzabile prima dell’incidente

- Osservabilità progettata per la ricostruzione

registrare input, contesto fornito al modello, versione e parametri di generazione: il minimo per spiegare, riprodurre, attribuire

- I log sono evidenza e bersaglio insieme

integrità: conservazione non riscrivibile, firma, catena di custodia

- La pipeline MLOps come catena di custodia degli artefatti

provenance dei dati, firma dei checkpoint, registry con versioni e rollback

- Auditabilità strutturale delle decisioni degli agenti

AI Act — art. 12

Regolamento (UE) 2024/1689:

per i sistemi ad alto rischio,
registrazione automatica degli
eventi lungo il ciclo di vita.

L’obbligo normativo converge
con la buona pratica di ingegneria.

Tre sfide aperte

Energia

Data center: 415 TWh nel 2024, oltre 945 TWh previsti al 2030 — più dell'intero Giappone (IEA, 2025).

L'ETTR ha una lettura energetica: tra 90% e 97% di tempo utile corrono gigawattora.

Evaluation

Il comportamento delle API commerciali cambia in mesi a parità di versione dichiarata (Chen, Zaharia, Zou, 2024).

Ogni valutazione ha una data di scadenza implicita; il regression testing comportamentale diventa necessità operativa.

Autonomia

Le «ironie dell'automazione» (Bainbridge, 1983): all'umano restano i casi peggiori, con meno pratica — e automation bias (Parasuraman & Riley, 1997).

Chi verifica l'AI che gestisce i sistemi?

Conclusioni: cinque tesi

- Il guasto è una condizione operativa permanente: la sincronia del training rovescia il failure domain
- La frontiera si è spostata dalla prevenzione alla velocità di ripristino

dal 90% al 97% di tempo utile in circa un anno di letteratura

- Nell'AIOps funziona ciò che incorpora il grounding su evidenze verificabili fin dal progetto
- Reliability e security condividono un unico spazio dei guasti

strumenti comuni, modelli di minaccia distinti: chaos engineering e red teaming

- L'autonomia si costruisce per gradi: reversibilità, privilegi minimi, auditabilità, benchmark realistici

Per la discussione in aula

- Se la telemetria fosse in parte attacker-controlled, come cambierebbe il vostro processo di incident response?
- Quali classi di azioni deleghereste per prime a un agente autonomo, e con quali guardrail tecnici e organizzativi?
- Come definireste uno SLO sulla qualità di un output non deterministico? Che cosa misurereste, e ogni quanto rivalutereste?
- L’ETTR come metrica di sostenibilità oltre che di affidabilità: ha senso portarla nel reporting ESG di un’organizzazione?

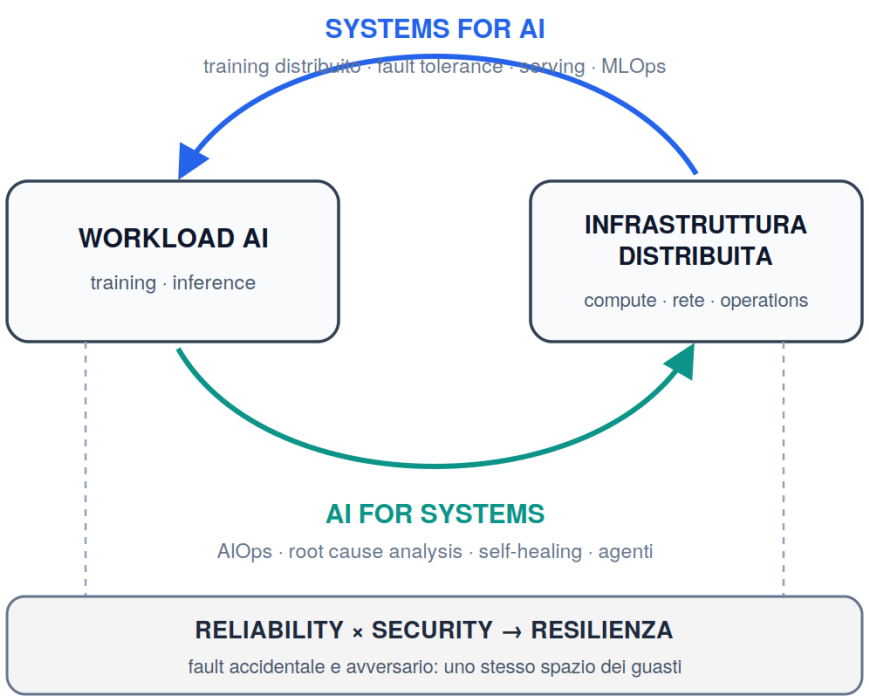
Riferimenti essenziali

Selezione dalla bibliografia completa del white paper (44 voci verificate)

- Avizienis et al., IEEE TDSC, 2004 — tassonomia della dependability
- Dean & Barroso, CACM, 2013 — The Tail at Scale
- Beyer et al., Site Reliability Engineering, O'Reilly, 2016
- Llama Team, AI @ Meta, 2024 — The Llama 3 Herd of Models
- Jiang et al., NSDI 2024 — MegaScale
- Wang et al., SOSP 2023 — Gemini (in-memory checkpointing)
- Wan et al., SOSP 2025 — ByteRobust
- Kwon et al., SOSP 2023 — vLLM / PagedAttention
- Zhong et al., OSDI 2024 — DistServe (goodput)
- Kreuzberger et al., IEEE Access, 2023 — MLOps
- Zhang et al., ACM Computing Surveys, 2025 — AIOps nell'era dei LLM
- Chen et al., EuroSys 2024 — RCACopilot
- Jha et al., ICML 2025 — ITBench · Chen et al., NeurIPS 2025 — STRATUS
- Vassilev et al., NIST AI 100-2 E2025 — adversarial ML
- Carlini et al., IEEE S&P 2024 — poisoning praticabile
- Greshake et al., AISec 2023 — indirect prompt injection · IEA, 2025

Scalable and Reliable Systems nell'era dell'AI

Architetture, operations e resilienza · sintesi del white paper



3 h
una interruzione imprevista ogni tre ore nel training di Llama 3 su 16.384 GPU
Meta, 2024

97%
Effective Training Time Ratio su un job di tre mesi a 9.600 GPU
ByteDance, SOSP 2025

11,4%
incidenti SRE risolti dagli agenti AI allo stato dell'arte
ITBench, ICML 2025

945 TWh
consumo elettrico dei data center previsto al 2030, dai 415 TWh del 2024
IEA, 2025

Fonti primarie verificate: NSDI · OSDI · SOSP · MLSys · ICML · NeurIPS · ACM Computing Surveys · IEEE S&P · NIST · IEA — Luglio 2026

Grazie · materiale didattico derivato dal white paper omonimo, luglio 2026