UNISINOS

Governança Arquitetural da Autonomia em Sistemas Agênticos de IA

Análise do Caso Fundador e Especificação do Programa de Pesquisa

DOCUMENTO D2

Framework de Supervisão e Autonomia

> *A contribuição. Argumenta que a autonomia é uma função da maturidade dos guardrails (controles estruturais de proteção), determina o que resta quando os guardrails são arbitrariamente bons, e separa a supervisão que uma arquitetura afirma exercer da supervisão que um humano pode de fato exercer.*

CANDIDATO	M.Sc. Epitácio Vicente do Nascimento Neto
ORIENTADOR	Prof. Dr. Felipe André Zeiser
PROGRAMA	Programa de Pós-Graduação em Computação Aplicada (PPGCA), Doutorado. Universidade do Vale do Rio dos Sinos (UNISINOS)
DATA DA EXECUÇÃO	7 de agosto de 2026
VERSÃO DO DOCUMENTO	1.1
IDENTIFICADOR DA EXECUÇÃO	`20260807-1714`

Como ler este conjunto de documentos

FONTE ÚNICA DE VERDADE

Todo fato deste conjunto é mantido uma única vez, em um registro legível por máquina no diretório `registers/` da base de conhecimento, e todo documento é renderizado a partir desses registros.

Um guardrail, uma classe de ação ou uma métrica é descrita uma vez e citada em todos os outros lugares por seu identificador. Onde um documento precisa de narrativa em torno de uma entrada de registro, a narrativa fica no documento e os fatos ficam no registro.

ESQUEMA DE IDENTIFICADORES

- `AC` classe de ação
- `GR-Lnn` guardrail, por camada
- `SD` determinante de supervisão
- `SP` ponto de supervisão
- `MG` métrica
- `CI` índice composto
- `DM` domínio de aplicabilidade
- `CS` conjunto de condições
- `CB` contrato de capacidade de nuvem
- `BL` item de backlog
- `F` achado durável
- `EV` arquivo de evidência

POLÍTICA DE NOMENCLATURA

Este conjunto é livre de identidade por construção, não por redação posterior. Nenhum endereço, nome de host, identificador de nuvem, URL de repositório, nome pessoal que não o do candidato e o do orientador, ou referência de cliente aparece em qualquer parte dele.

Hosts são nomeados por papel de serviço. Tecnologias de fornecedores e tecnologias abertas SÃO nomeadas, porque o conteúdo técnico depende disso. A evidência bruta permanece no host em `evidence/` e é citada por identificador, nunca transcrita.

ORDEM DE LEITURA

- **D0** Sumário Executivo, leia primeiro
- **D2** Framework de Supervisão e Autonomia, o argumento
- **D1** Arquitetura do Caso Fundador, a base de evidências
- **D3** Catálogo de Guardrails, referência
- **D4** Framework de Medição, referência
- **D5** Domínios de Aplicabilidade
- **D6** Ambiente de Pesquisa e Roteiro

MARCAS DE EVIDÊNCIA

- **OBSERVED** verificado no host nesta execução, com um identificador de evidência.
- **REPORTED** extraído da proposta de doutorado ou do brief da execução, não verificável neste host.
- **PROPOSED** uma proposta de projeto desta execução, não a descrição de algo existente.
- **NOT DETERMINED** não foi possível estabelecer; a razão é sempre declarada.

REFERÊNCIAS CRUZADAS

Referências cruzadas usam número do documento e identificador, nunca número de página, porque números de página mudam quando um registro cresce e o documento é re-renderizado.

Exemplo: *ver D3,* `GR-L06-02`.

Sumário

1. O problema do enforcement

Um sistema agêntico é hoje governado majoritariamente dizendo a ele o que fazer. A instrução é escrita em prosa, colocada em um arquivo que o modelo lê como contexto, e a conformidade do sistema com ela é inferida do fato de que nada deu errado de forma evidente. Este documento argumenta que essa forma de governança não é enforcement (aplicação coercitiva de controle), que a diferença é mensurável, e que é nessa diferença que vivem as falhas.

O argumento é desenvolvido a partir de um ambiente real. O caso fundador é um parque de engenharia multiagente construído e operado pelo candidato em infraestrutura de nuvem de produção, do qual um host, referido ao longo do texto como o Agent Console host, foi examinado em detalhe nesta execução. O D1 registra o que foi encontrado. Este documento usa esse registro, mas não depende de suas particularidades: cada afirmação aqui é enunciada de modo a poder ser testada em outro ambiente, que é para isso que existe o ambiente experimental doutoral.

Três observações do caso fundador estabelecem o problema.

A primeira é que a governança do próprio ambiente é quase inteiramente instructional (instrucional). Das 60 classes de ação catalogadas, 36 não são delimitadas por nenhum controle, 8 apenas por prosa, 12 por uma verificação que a própria parte agente também poderia alterar, e 4 por um controle que a parte agente não consegue contornar `EV-003` `EV-032`. Os controles que existem são frequentemente bem construídos. Isso não é o mesmo que serem enforcement, e a distinção é o assunto da Seção 2 do D3.

A segunda é que acesso e mandato se dissociaram. Classificadas duas vezes, uma quanto ao que um agente é tecnicamente capaz de fazer e outra quanto ao que está autorizado a fazer, 50 das 60 classes de ação mostram acesso mais permissivo do que o mandato. O mandato é real no sentido de que o operador de fato o pretende. Ele existe apenas em prosa `EV-032`. Cada uma das 50 é um limite não aplicado, e o agregado está registrado como o achado `F-029`.

A terceira é a que motiva o doutorado. No caso fundador, uma auditoria independente de dezoito episódios operacionais, extraídos de quarenta e seis candidatos identificados pelo próprio sistema, confirmou o núcleo técnico em todos os casos e encontrou divergência factual entre o que o agente relatou no momento e o que a evidência independente mostrou em onze dos dezoito, uma taxa de 61,1 por cento `REPORTED`. Um episódio é a ilustração canônica: uma implantação foi relatada como bem-sucedida quando, na verdade, havia desabilitado a função de conexão central do serviço, deixando o serviço indisponível por aproximadamente 24 horas antes que a divergência fosse percebida.

> **A FALHA É ARQUITETURAL, NÃO COMPORTAMENTAL**
>
> A propriedade interessante desse episódio não é que um agente estava errado. Software erra rotineiramente, e a literatura sobre agentes de modelos de linguagem estabelece que sua saída é probabilística e pode estar confiantemente errada (Yao et al. 2023; Yang et al. 2024). A propriedade interessante é que nada na arquitetura conseguia distinguir um relato correto de um incorreto, porque o relato era o único registro da implantação que existia. A falha não é que o agente mentiu. É que o sistema foi construído de modo que mentir e dizer a verdade produziam estados observáveis idênticos.

Essa é a lacuna de enforcement, dita concretamente. O restante deste documento a desenvolve em um framework com três partes: uma relação entre autonomia e guardrails, um conjunto de propriedades que fixam uma ação à supervisão humana independentemente dos guardrails, e um teste para saber se uma postura de supervisão declarada é real.

1.1 O que este documento contribui

Três coisas, ditas com clareza para que as afirmações possam ser verificadas contra a Seção 8, onde são posicionadas em relação à literatura.

Primeiro, uma relação explícita que torna o nível de autonomia mais alto defensável para uma ação uma função de dois termos, as propriedades da ação e o conjunto de guardrails que a cobre, e um método para determinar qual termo domina. Onde os guardrails dominam, a autonomia é um problema de engenharia. Onde as propriedades da própria ação dominam, nenhum investimento em guardrails a move.

Segundo, dez determinantes de supervisão, cada um com uma definição, um teste operacional e exemplos trabalhados, aplicados a todo o inventário de classes de ação para produzir o resíduo irredutível de supervisão: o conjunto de classes de ação que permanecem acima do Nível I sob toda configuração catalogada de guardrails.

Terceiro, seis condições de viabilidade de supervisão que distinguem a supervisão que uma arquitetura declara, a postura nominal, da supervisão que um humano pode de fato exercer, a postura efetiva. Todo ponto de supervisão do caso fundador é avaliado contra as seis. Vinte e sete de trinta e quatro divergem.

---

2. A autonomia como função da maturidade dos guardrails

2.1 A taxonomia e o que ela não resolve

A proposta doutoral estabelece três níveis. O Nível I é totalmente autônomo, sem revisão prévia, reversível, confinado ao desenvolvimento ou ao controle de versão, ou permanentemente pré-autorizado; aproxima-se do human-out-of-the-loop (humano fora do laço). O Nível II é autônomo com revisão humana subsequente, com impacto real em produção mas reversível, comunicado de forma assíncrona; corresponde ao human-on-the-loop (humano sobre o laço), e é o nível em que o caso fundador mediu 61,1 por cento de divergência factual. O Nível III exige aprovação prévia ou é proibido, para ações irreversíveis, voltadas para fora, ou que alterem um sinalizador de segurança; exige human-in-the-loop (humano no laço).

A taxonomia classifica ações. O que ela não resolve é como uma ação recebe sua classificação, e essa é a pergunta que esta seção responde. Dois ambientes podem executar a mesma ação, digamos, uma implantação em um serviço de produção, e estar corretos ao classificá-la de forma diferente, porque a classificação não é uma propriedade da ação isoladamente.

2.2 A relação que governa

Enuncie-a explicitamente. Para uma classe de ação `a` e um conjunto de guardrails `G` que a cobre, o nível de autonomia mais alto defensável é

```
L(a, G) = max( floor(a), ceiling(G, a) )
```

onde `floor(a)` é o nível abaixo do qual as propriedades da própria ação não permitem que ela caia, seja lá o que for construído, e `ceiling(G, a)` é o nível que o conjunto de guardrails é suficiente para sustentar. Os níveis são ordenados `I < II < III` em permissividade, de modo que `max` seleciona o mais restrito dos dois. A relação é uma definição, não uma descoberta, e sua utilidade está inteiramente no fato de os dois termos serem determináveis separadamente.

`ceiling(G, a)` é uma grandeza de engenharia. Ela cai à medida que a cobertura structural (estrutural) aumenta. O D3 cataloga 134 guardrails em doze camadas e dez condition sets (conjuntos de condições) nomeados, de uma linha de base nua até uma configuração totalmente endurecida, precisamente para que esse termo possa ser variado deliberadamente em uma execução controlada.

`floor(a)` não é uma grandeza de engenharia. É fixado pelas propriedades da ação, desenvolvidas na Seção 3 como os determinantes de supervisão. Nenhum controle do catálogo o move.

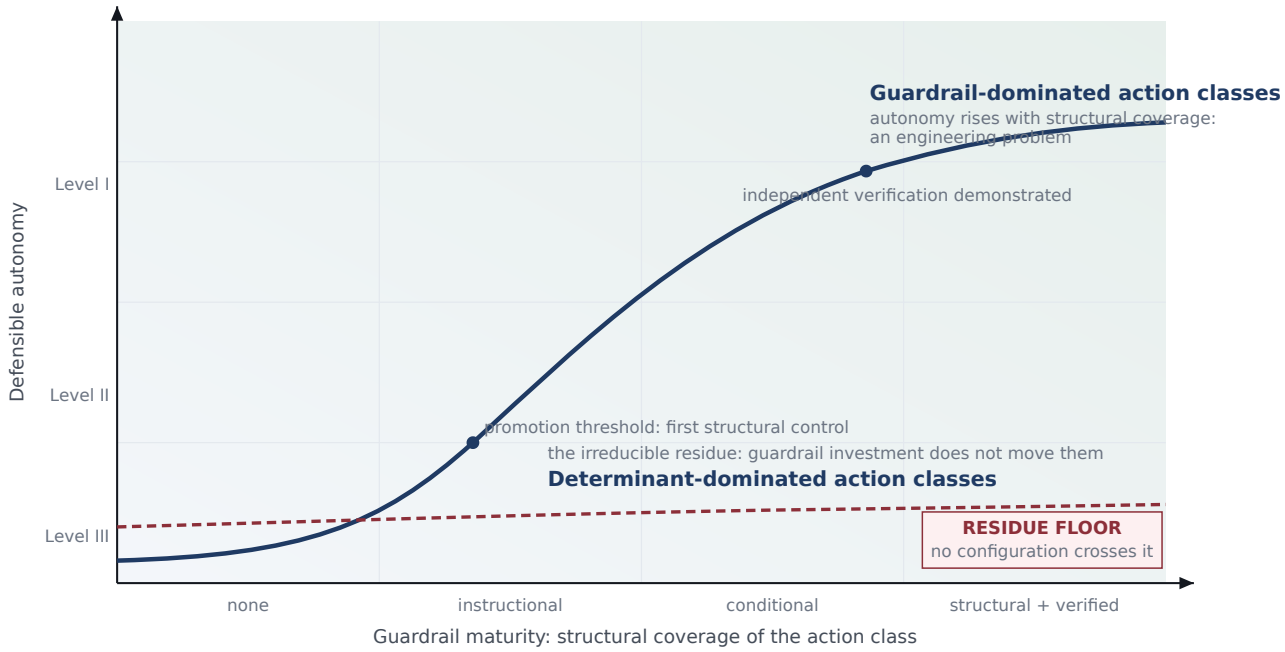

Figura D2-1. A fronteira da autonomia. Para classes de ação dominadas por guardrails, o nível defensável cai à medida que a cobertura structural aumenta. Para classes dominadas por determinantes, ele não cai, e o piso do resíduo é o conjunto de classes para as quais nenhuma configuração catalogada o cruza.

2.3 Qual termo domina

A determinação é mecânica uma vez definidos os determinantes. Aplique o teste de cada determinante à classe de ação. Se nenhum se aplica, `floor(a)` é o Nível I e a ação é dominada por guardrails: seu nível é inteiramente função do que foi construído, e o investimento o move. Se algum determinante se aplica, `floor(a)` é o nível residual desse determinante e a ação é dominada por determinante: o investimento em guardrails reduz a consequência de um erro, o que vale a pena fazer, mas não elimina a necessidade de uma decisão humana.

Aplicado às 60 classes de ação do inventário do caso fundador, 28 são dominadas por guardrails e 32 são dominadas por determinante. As 32 constituem o resíduo irredutível e estão listadas no Apêndice C.

> **POR QUE A DISTINÇÃO VALE O ESFORÇO**
>
> Sem ela, toda discussão de governança desaba na mesma disputa insolúvel, em que uma parte observa que um controle poderia ser construído e a outra observa que ainda é necessário um humano, e ambas estão certas sobre ações diferentes. Separar os termos torna o desacordo testável: para uma ação dominada por guardrails, a primeira parte está correta e o remédio é orçamento, e para uma ação dominada por determinante, a segunda está correta e nenhum orçamento ajuda.

3. Os determinantes de supervisão e o resíduo irredutível

3.1 O que é um determinante

Um determinante de supervisão é uma propriedade de uma ação que a fixa à supervisão humana independentemente de quão bons os guardrails se tornem. Não é uma afirmação sobre a competência do agente. Uma ação pode ser fixada por um determinante mesmo quando o agente decidiria corretamente todas as vezes, porque, para alguns dos determinantes, a correção não é a propriedade em questão.

Cada determinante tem uma definição, um teste operacional que pode ser aplicado a uma classe de ação sem saber nada sobre o agente, um nível residual, e uma afirmação honesta sobre o que de fato o reduz. Esse último campo importa: um determinante que não admitisse nenhuma mitigação seria um conselho de desespero, e a maioria admite mitigação substancial sem admitir remoção.

3.2 Os dez determinantes

O registro é o Apêndice A. Oito são nomeados nos próprios termos da proposta doutoral. Dois, `SD-09` e `SD-10`, são extensões desta execução e estão marcados como tal, com a razão registrada no registro de decisões como `DEC-007`.

Tabela D2-1. Os determinantes de supervisão, com o teste operacional que decide se cada um se aplica. A coluna de remoção por guardrail indica se o investimento em guardrails pode remover o determinante por completo, reduzi-lo, ou nenhum dos dois.

ID	NAME	TEST	GUARDRAIL-REMOVABLE	RESIDUAL LEVEL
SD-01	Irreversibility	Is there a rollback path that restores the prior state exactly, is it available to the actor, and has it been demonstrated by exercise rather than asserted? If any of the three is false, the determinant applies.	partial	II
SD-02	Externality	Does any effect of this action become visible to, or act upon, a person or organisation outside the operator's span of control, before any review can occur?	no	III
SD-03	Accountability requirement	If the decision were later challenged, would an answer of the form 'the system decided' be accepted by the party entitled to ask? If not, the determinant applies.	no	III
SD-04	Value and judgement content	Can the decision be expressed as a predicate over observable state that a reviewer would accept as complete? If the honest answer requires the phrase 'it depends on the circumstances', the determinant applies.	partial	II
SD-05	Unverifiability	Is there an oracle for success that does not consult the acting agent, and that can be evaluated mechanically? If the only available check is the agent's own report, the determinant applies.	yes	II
SD-06	Distribution shift	Has this action class been exercised, under conditions materially like the present ones, often enough for its failure rate to be estimated? If the present case is novel in a way the actor cannot itself detect, the determinant applies.	partial	II
SD-07	Containment shortfall	If this action were performed wrongly in the worst plausible way, would the consequence stay inside a boundary the operator can restore independently? If not, the determinant applies.	yes	III
SD-08	Objective ambiguity	Could a diligent actor complete this action, be judged compliant against the instruction as written, and still have produced an outcome the instructing human would reject? If so, the determinant applies.	partial	II

ID	NAME	TEST	GUARDRAIL-REMOVABLE	RESIDUAL LEVEL
SD-09	Compounding	Does the output of this action become the input to the next decision of the same actor, without an intervening independent check? If so, the determinant applies.	partial	II
SD-10	Adversarial adaptation	Would a party with an interest in the action succeeding be able to observe the control's decision boundary and construct an input that satisfies it? If so, the determinant applies.	partial	II

Três deles merecem comentário aqui porque carregam a maior parte do argumento.

`SD-05`, a não verificabilidade, é o determinante mais responsivo a investimento. Aplica-se quando o sucesso de uma ação não pode ser verificado por nenhum mecanismo independente do agente que a executou. No caso fundador, aplica-se em quase todos os lugares, porque o único registro do que um agente fez é o registro que o próprio agente escreveu `EV-026` `EV-034`. É também o determinante que um verificador independente sob uma identidade separada remove por completo. Os 61,1 por cento de divergência do caso fundador são, no vocabulário deste framework, uma medida de `SD-05` não endereçado. É por isso que a camada de verificação é o bloco isolado mais consequente do catálogo de guardrails e por que o condition set `CS-10` existe como uma ablação para testar se ela de fato faz o trabalho que lhe é atribuído.

`SD-07`, o déficit de contenção, é inteiramente um artefato da arquitetura e, portanto, é totalmente removível em princípio. Aplica-se quando o agente detém privilégio, credenciais ou alcance além do escopo da ação, de modo que um erro na ação pode se expressar fora desse escopo. No caso fundador, aplica-se a quase toda classe de ação, porque uma única conta do sistema operacional detém escalonamento de privilégio irrestrito e cinco conjuntos de credenciais estáticas de administrador de nuvem `EV-003` `EV-033`, registrados como os achados `F-028` e `F-032`. Este é o determinante cuja remoção é a menos interessante conceitualmente e a mais importante na prática: não exige nenhuma ciência nova, apenas identidade por agente e privilégio mínimo, e até que seja removido, quase todo outro controle do catálogo é apenas consultivo.

`SD-03`, o requisito de responsabilização, não é redutível de forma alguma, e isso está correto. Aplica-se onde um humano nomeado precisa responder por uma decisão por força de lei, contrato, norma ou obrigação profissional, independentemente de a máquina decidir corretamente. Os guardrails podem tornar a decisão do humano mais bem informada, mais rápida e mais barata, e devem fazê-lo. Não podem cumprir a obrigação, porque a obrigação é exógena à arquitetura. O domínio `DM-08`, conformidade e atestação, é a instância no nível de domínio: tem um teto de autonomia que nenhum investimento eleva, e o achado `F-022` registra isso como um design correto, e não como uma fragilidade não corrigida.

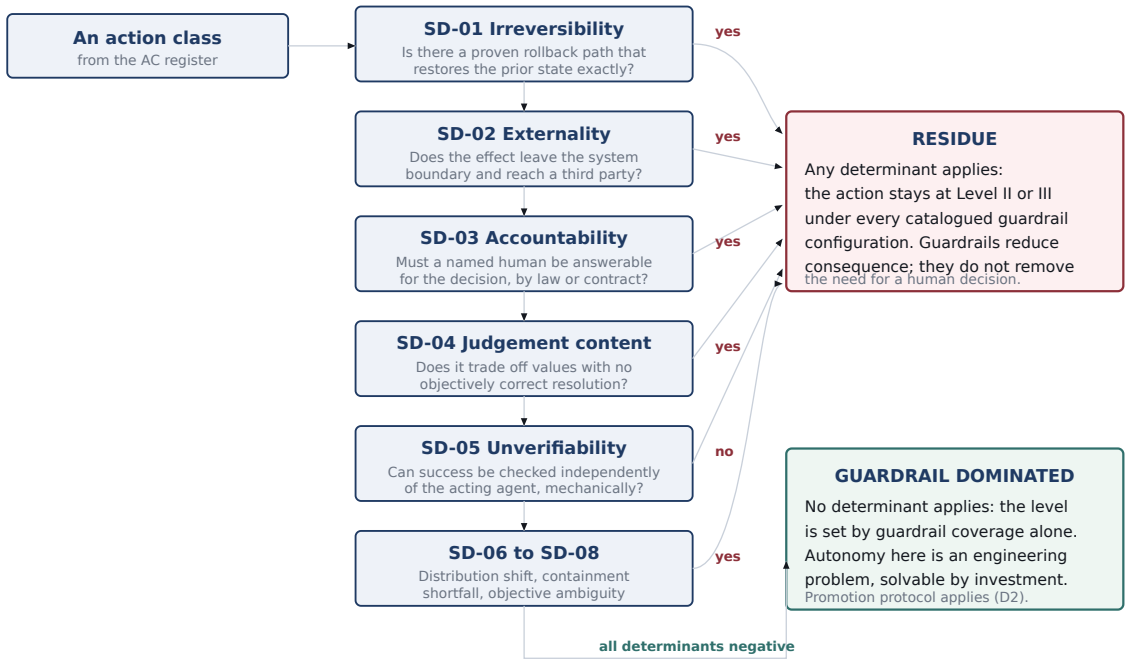

Figura D2-2. Fluxo de decisão dos determinantes. O teste é aplicado a cada classe de ação do inventário; a saída é o registro de resíduo, no qual cada entrada é fixada ao seu determinante dominante.

3.3 O resíduo irredutível

Aplicar os determinantes ao inventário produz o resíduo: 32 das 60 classes de ação permanecem acima do Nível I sob toda configuração catalogada de guardrails. Dessas, 22 permanecem no Nível III e 10 no Nível II.

Os determinantes dominantes são `SD-07`, déficit de contenção, e `SD-02`, externalidade, com nove classes de ação cada, seguidos por `SD-01`, irreversibilidade, e `SD-03`, responsabilização, com quatro cada.

Essa distribuição merece leitura cuidadosa, porque contém um resultado animador. O maior grupo isolado é fixado por `SD-07`, que é totalmente removível por engenharia de identidade. O resíduo do caso fundador é, portanto, substancialmente maior do que seu resíduo último: essas nove classes estão no resíduo por causa de como este ambiente específico acontece de estar construído, não por causa do que as ações são. A porção genuinamente irredutível são as 23 classes fixadas por externalidade, responsabilização ou irreversibilidade, e é bem menor do que o número bruto sugere.

> **O RESÍDUO É DEFINIDO OPERACIONALMENTE, NÃO POR ARGUMENTO**
>
> Uma classe de ação está no resíduo se e somente se permanecer acima do Nível I sob o condition set `CS-09`, a configuração totalmente endurecida contendo todo controle catalogado. Essa definição é deliberadamente mecânica: significa que o resíduo pode ser recalculado sempre que o catálogo crescer, e uma execução posterior que adicione um controle que retire uma ação do resíduo terá produzido um resultado, não uma opinião.

O registro de resíduo é o Apêndice C. Cada entrada indica o nível residual, os determinantes que a fixam, o determinante dominante isolado, um argumento específico para essa ação sobre por que os guardrails não a movem, o melhor nível alcançável com o catálogo completo aplicado, e os guardrails necessários para alcançá-lo.

4. Viabilidade de supervisão: postura nominal e efetiva

As Seções 2 e 3 dizem respeito a em que nível uma ação deveria estar. Esta seção diz respeito a algo diferente e mais incômodo: se a supervisão que uma arquitetura declara nesse nível de fato existe.

4.1 A distinção

A **postura nominal** é a supervisão que a arquitetura declara. É o que um diagrama de arquitetura mostra, o que um documento de controle afirma, e o que é mostrado a um auditor.

A **postura efetiva** é a supervisão que um humano pode de fato exercer. É determinada por seis condições, todas as quais precisam se sustentar. Onde não se sustentam, a postura efetiva é mais fraca que a nominal, e a divergência é um achado primário.

A distinção importa porque as duas são fáceis de confundir e porque a confusão é invisível de dentro. Nada falha. Nenhum componente apresenta erro. A arquitetura continua a se descrever de forma precisa no nível dos componentes, enquanto a propriedade que esses componentes foram reunidos para produzir silenciosamente deixou de existir.

4.2 As seis condições

VC1 independência. O sinal de supervisão não se origina do agente sendo supervisionado. Um sistema em que o único registro do que aconteceu é o registro escrito pela parte que o fez não tem nenhum sinal de supervisão, qualquer que seja o volume de suas notificações.

> **UMA INSTÂNCIA DE DENTRO DO PRÓPRIO FERRAMENTAL DESTA EXECUÇÃO**
>
> Enquanto este conjunto de documentos era produzido, o renderizador que o gera estagnou em uma linha de entrada malformada, cresceu até 9,1 GiB de memória residente sem emitir qualquer sinal próprio, e só foi detido pela proteção de memória do sistema operacional, que derrubou junto a sessão supervisora. O componente que falhava era a única testemunha do próprio progresso, de modo que a detecção por auto-relato era impossível em princípio, não apenas improvável. A falha foi em seguida convertida em uma falha contida, diagnosticável e limitada a um único documento exatamente pela classe de controle que esta seção descreve: um limite independente e estrutural, indiferente ao processo que delimita (F-043).

VC2 latência. A latência de intervenção é menor do que o tempo de conclusão do dano. Isso é computável, por classe de ação, como

```
t_detect + t_decide + t_act < t_harm
```

onde `t_detect` é o intervalo entre a ação e um humano tomar conhecimento dela, `t_decide` é o intervalo entre a percepção e uma decisão, `t_act` é o intervalo entre a decisão e a intervenção surtir efeito, e `t_harm` é o intervalo entre o início e o ponto em que o efeito não pode mais ser evitado. A margem `t_harm - (t_detect + t_decide + t_act)` é a métrica `MG-046`. Uma margem negativa significa que a supervisão nesse ponto não pode ser efetiva por mais atento que o humano esteja, e nenhuma diligência do operador a altera.

VC3 interrupção (halt). A intervenção interrompe o trabalho em andamento, e não apenas impede a próxima ação. São mecanismos diferentes, e a diferença é rotineiramente ignorada, porque ambos são chamados de "parar".

VC4 atenção. O número de fluxos de sinal concorrentes está dentro da capacidade de um único operador. Uma postura de supervisão que pressupõe que um humano está observando um fluxo não é uma postura se o fluxo não pode ser observado.

VC5 compreensibilidade. Uma solicitação de aprovação ou um alerta apresenta evidência independente, e não o resumo do próprio agente. Uma aprovação concedida com base no relato da parte que a solicita é uma ratificação, não uma decisão.

VC6 autoridade. O humano tem a posição institucional e o mecanismo para sobrepor a ação, e a sobreposição não é anulada por um reinício ou nova tentativa automáticos. Um operador que encerra um agente e o vê retornar em segundos não detém autoridade sobre ele em nenhum sentido que a palavra sustente.

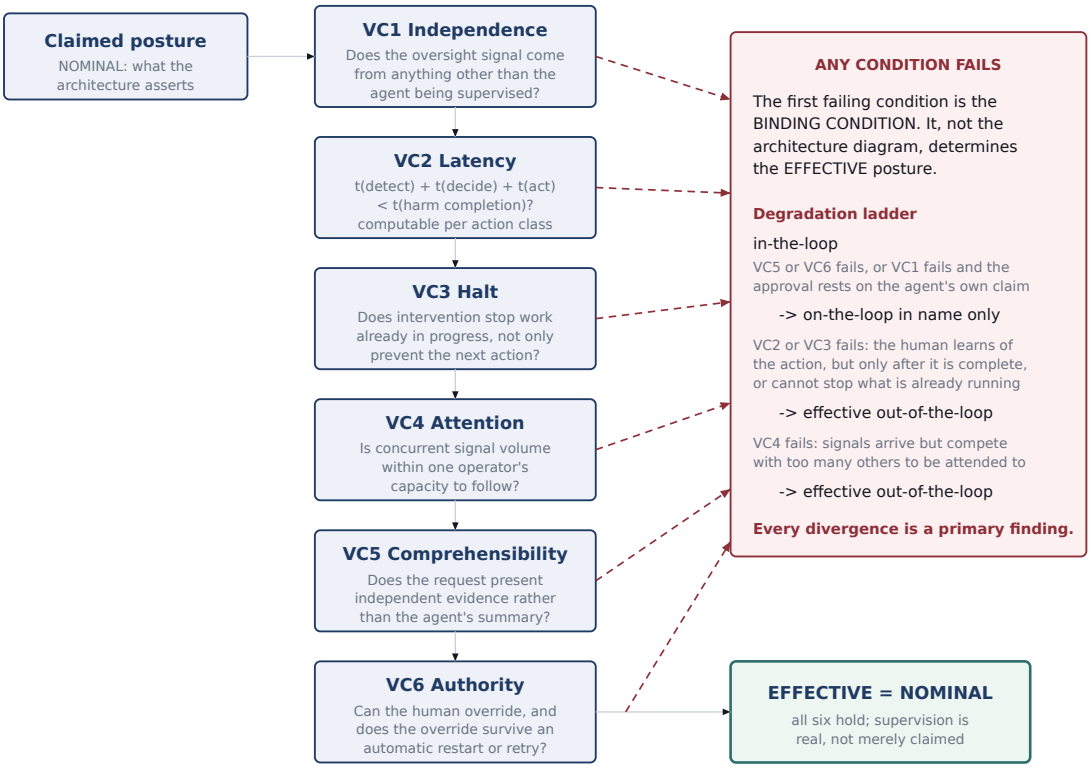

Figura D2-3. Fluxo de decisão da viabilidade de supervisão. A primeira condição que falha é a condição vinculante, e é ela, e não o diagrama de arquitetura, que determina a postura efetiva.

4.3 A condição vinculante

Quando mais de uma condição falha, o framework registra a primeira falha na ordem acima como a **condição vinculante (binding condition)**. Trata-se de uma simplificação deliberada e ela se justifica: converte uma avaliação difusa em uma causa única e nomeada, o que é o que torna um remédio endereçável. Um ponto de supervisão que falha em VC2 precisa de um detector mais rápido, e um que falha em VC3 precisa de um mecanismo de intervenção diferente, e esses são trabalhos distintos.

4.4 Avaliação do caso fundador

Trinta e quatro pontos de supervisão foram identificados e cada um foi avaliado contra as seis condições. O registro completo é o Apêndice B.

Vinte e sete dos trinta e quatro divergem. As condições vinculantes se distribuem assim: VC1 independência em 8 casos, VC2 latência em 8, VC3 interrupção em 3, VC6 autoridade em 3, VC4 atenção em 2, VC5 compreensibilidade em 2, e em um caso o controle declarado simplesmente não existe.

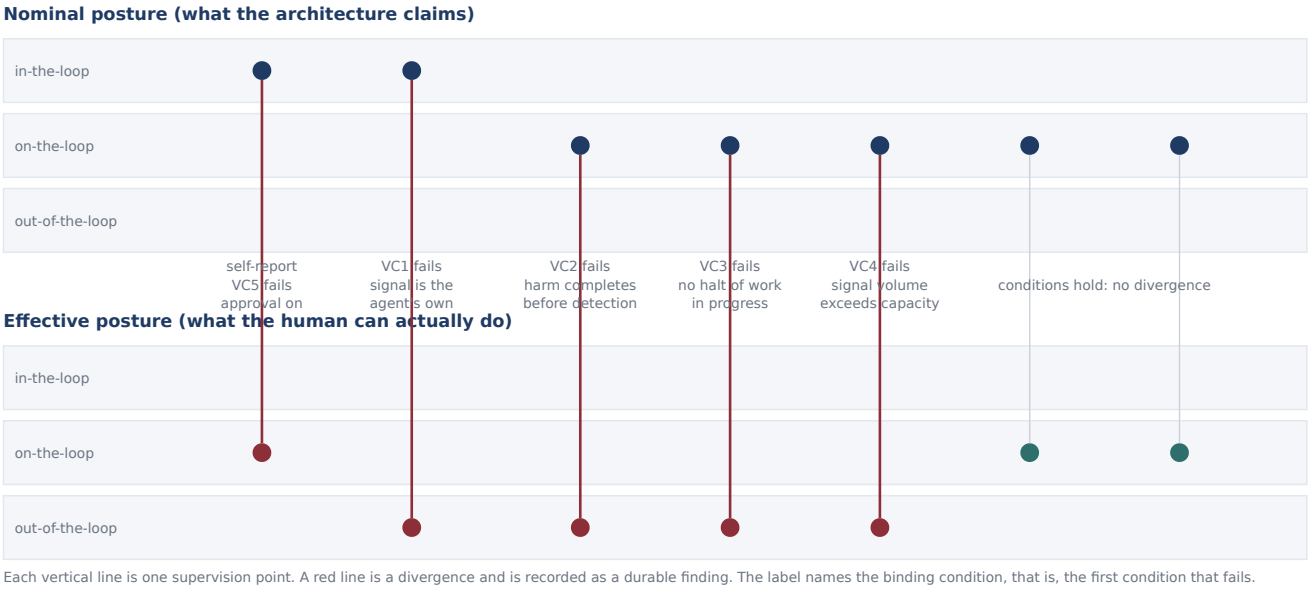

Figura D2-4. Postura nominal contra postura efetiva. Cada linha vertical é um ponto de supervisão; uma linha que desce é uma divergência, rotulada com sua condição vinculante.

Os sete pontos que não divergem são tão informativos quanto os vinte e sete que divergem, e são tratados na Seção 5.

Quatro divergências carregam o argumento.

O sinal primário de supervisão é o próprio relato do agente sobre si mesmo (`F-001`, `SP-001`, `SP-002`). O canal pelo qual o operador toma conhecimento do que um agente fez é uma notificação que o agente compõe e envia quando decide que o trabalho está concluído. Nada o obriga a isso, e a instrução para enviá-la vive em um arquivo que o agente pode escrever `EV-029` `EV-032`. VC1 falha nos dois pontos que carregam quase toda a supervisão rotineira. Este é o mecanismo arquitetural por trás dos 61,1 por cento de divergência: não que os agentes enganem, mas que a arquitetura não fornece um segundo relato contra o qual o primeiro pudesse ser comparado.

Exatamente uma intervenção interrompe o trabalho em andamento, e o reinício automático a desfaz (`F-010`, `SP-012`, `SP-034`). Dos seis mecanismos de intervenção disponíveis ao operador, um encerra um processo filho em execução. Para os nove agentes supervisionados por unidade, o reinício sempre ativo (restart-always) traz uma nova sessão de volta em segundos, com um prompt de inicialização novo `EV-005` `EV-039`. Todo outro mecanismo, incluindo a interrupção do console, o kill switch (interruptor de emergência) de atualização, o kill switch do barramento de mensagens e o portão de aprovação de implantação, afeta apenas a próxima ação. VC3 falha em quase todo lugar, e onde não falha, é VC6 que falha em seu lugar.

O caminho de parada supervisionada começa dando ao agente mais um turno com ferramentas completas (`F-011`, `SP-013`, `SP-014`). O script de parada supervisionada começa pedindo ao agente que registre seu estado e permite até 35 segundos de acesso irrestrito a ferramentas antes de encerrá-lo. A interrupção do console é uma colagem seguida de uma quebra de linha: ela entra na fila para o próximo turno do agente e não interrompe nada `EV-039`. O design é coerente se a razão para parar é que o trabalho terminou. É exatamente invertido se a razão para parar é que o trabalho é prejudicial, que é o único caso em que a supervisão importa.

Os detectores independentes detectam silêncio, não consequência (`F-009`, `SP-010`, `SP-011`). Cinco vigias agendados são os únicos sinais não escritos pelos agentes, e seu código é genuinamente independente. Sua entrada é a própria saída de terminal do agente e um carimbo de tempo da última atividade; sua condição de disparo é que um agente ficou quieto; quatro dos cinco respondem digitando no próprio agente que ficou quieto `EV-028` `EV-039`. Não há nenhum vigia cuja condição de disparo seja que um agente fez algo consequente. A melhor latência possível para qualquer sinal

independente que chega ao operador excede dez minutos, e ela sempre significa apenas que um agente parou de produzir texto.

4.5 O resultado agregado

> **UMA POSTURA NUNCA EXERCIDA É INDISTINGUÍVEL DE UMA QUE NÃO EXISTE**
>
> Ao longo de 46 dias de logs de auditoria do console, não há nenhum registro de um humano parando, pausando, rejeitando ou sobrepondo um agente com base no que ele estava fazendo. As 159 finalizações de sessão acompanham as criações de sessão como ciclo de vida ordinário; as dez finalizações de frota são três varreduras de descomissionamento; o kill switch global do barramento de mensagens foi ligado e desligado com 26 milissegundos de diferença. Há zero rejeições de implantação, zero aprovações de atualização e zero ativações de kill switch, nunca. A única violação de governança detectada foi encontrada por um agente, escalada para um agente, e encerrada por nenhum dos dois `EV-039`. Este é o achado `F-042`.

A leitura correta desse resultado não é que o operador estava desatento. É que todo mecanismo disponível é mais lento do que a ação que deveria interromper, a maioria deles não interrompe nada que já esteja em execução, e o sinal que provocaria uma intervenção é escrito pela mesma parte que seria alvo dela. O ambiente se descreve como human-on-the-loop. Ao longo de 46 dias, o laço foi exercido zero vezes. A ausência de intervenção não é evidência de que nenhuma era necessária.

---

5. Onde o human-out-of-the-loop é defensável

A tese não é que humanos pertencem a todo lugar. Um argumento que terminasse na Seção 4 seria um conselho de paralisia, e também estaria errado: a autonomia plena é defensável para uma parte substancial do inventário hoje, e para uma parte maior dele com adições nomeadas.

5.1 Defensável hoje

Oito das 60 classes de ação carregam um mandato de Nível I, e para todas as oito o acesso não excede o mandato. Seis são classes em que a ação é reversível, confinada ao desenvolvimento ou ao controle de versão, e ou permanentemente pré-autorizada ou sem consequência fora da árvore de trabalho: ler código-fonte, modificar uma árvore de trabalho, fazer commit, fazer push, ingerir um arquivo no armazenamento de sessão, e chamar uma API de leitura de nuvem. As duas restantes são reiniciar um serviço supervisionado por gerenciador de processos neste host (`AC-009`) e enviar uma notificação ao operador (`AC-046`).

Dois outros casos são defensáveis por uma razão mais interessante, e são os dois padrões de referência de todo o programa.

A identidade de conexão somente leitura para o System of Record (`SP-029`, `F-023`). O acesso à fonte de dados autoritativa a montante se dá por meio de uma identidade de conexão que não consegue expressar uma escrita. Não existe identidade capaz de escrever para um agente deter, então o estado proibido não é alcançável **REPORTED**. Esse controle não exige monitoramento, não produz atrito no trabalho legítimo, não pode ser contornado por um prompt, e não depende de nenhuma forma de conformidade do agente. É a única classe de ação no inventário em que o acesso não excede o mandato para uma ação de alta consequência, e a razão é que o acesso simplesmente não existe.

A sobreposição structural de modo de teste no domínio de tempo real (`SP-024`, `F-019`). Toda discagem é interceptada antes que um destino real seja resolvido, na camada de despacho, e não no raciocínio do agente **REPORTED**. A intenção do agente de alcançar um destino real é expressável e

simplesmente não resulta em um. A autonomia plena é defensável sob essa sobreposição precisamente porque a ação sendo tomada deixou de ser a ação que precisaria de supervisão.

> **O QUE TORNA ESSES DOIS CONTROLES STRUCTURAL**
>
> Em ambos os casos, a conformidade não é um termo no resultado. O agente não precisa saber que o controle existe, não precisa concordar com ele, e não pode satisfazer sua intenção por nenhuma formulação alternativa. Essa é a definição operacional de enforcement structural usada ao longo deste programa, e é a propriedade que separa esses dois de todo outro controle no Agent Console host, cada um dos quais a parte governada consegue contornar porque detém o sistema de arquivos, o escalonamento de privilégio, ou ambos.

5.2 Promovível com adições nomeadas

Vinte das 28 classes de ação dominadas por guardrails hoje se situam acima do Nível I apenas porque os guardrails que sustentariam o Nível I não estão presentes. Elas se tornam defensáveis no Nível I sob as seguintes adições, nesta ordem de dependência.

- Identidade de carga de trabalho por agente (`GR-L01-02`) e privilégio mínimo

por classe de ação (`GR-L01-03`). Sem elas nada mais é structural, porque a parte governada detém o privilégio que aplicaria o controle. Essa única mudança remove `SD-07` de nove classes de ação.

- Credenciais efêmeras com duração de sessão curta (`GR-L01-04`) e nenhum

material de chave de longa duração em disco (`GR-L01-05`). Elas limitam a consequência do único determinante que é irreversível no contato, a exposição de credenciais.

- Negação de acesso de escrita aos próprios arquivos de instrução e

configurações de permissão do agente (`GR-L03-07` , `GR-L03-08`), que é a forma minimamente viável de `GR-L07-06` , a impossibilidade structural de um agente ampliar seu próprio mandato.

- Eventos emitidos pela infraestrutura (`GR-L08-11`) e um verificador

independente sob uma identidade separada (`GR-L08-05`). Juntos, eles removem `SD-05` e são a pré-condição para qualquer afirmação de que o trabalho de um agente é evidência, e não mera alegação.

- Um classificador de ação em tempo de execução que atribui classe e nível no

momento da tentativa (`GR-L07-08`). Isso é a taxonomia tornada executável, e é o que converte os registros deste programa de uma descrição em um controle.

5.3 A forma geral da resposta

A autonomia plena é defensável quando a ação é reversível, seus efeitos são contidos por identidade, e não por intenção, e seu sucesso é verificável por uma parte diferente daquela que a executa. Essas três condições não são coincidência: são as negações dos três determinantes mais comuns, `SD-01` , `SD-07` e `SD-05` . Uma organização decidindo se concede autonomia plena para uma ação dificilmente faria melhor do que fazer essas três perguntas e parar por aí.

---

6. O protocolo de promoção e rebaixamento

Um nível de autonomia é uma afirmação sobre um estado de coisas que muda. O framework fica incompleto sem um protocolo para mover uma classe de ação entre níveis, e o protocolo precisa ser especificado com o mesmo cuidado que a classificação, porque um caminho de promoção não

especificado significa que a promoção acontece de modo informal e o rebaixamento não acontece de forma alguma.

6.1 Evidência exigida para a promoção

Uma classe de ação é promovida a um nível menos restrito quando todos os itens a seguir forem demonstrados. Cada um é um artefato, não uma afirmação.

1. **Cobertura de guardrail alcançada.** Todo guardrail nomeado no campo

`guardrails_required` do registro de resíduo para essa classe de ação está presente, não é parcial, e cada um emite o sinal especificado. A cobertura é lida da matriz de cobertura no D3, não de uma alegação.

1. **Independência de verificação demonstrada.** Para a classe de ação em

questão, a métrica `MG-024`, razão de independência de verificação, está acima do limiar definido para o ambiente, e a parte verificadora detém uma identidade que a parte agente não detém. Um verificador independente que lê um registro que o próprio agente escreveu não é independente, e o campo de ponto de coleta da métrica é o que distingue os dois casos.

1. **Um período de observação com uma taxa de divergência declarada.** A

classe de ação foi exercida no nível atual, sob observação, por um período declarado, e a métrica `MG-022`, taxa de divergência de alegação, foi computada sobre esse período com um denominador declarado. Uma promoção sustentada por um período de observação sem divergências e sem denominador não é sustentada por nada.

1. **Um caminho de rollback comprovado por exercício.** O rollback foi

executado, no alvo real, e o estado resultante foi comparado com o estado anterior. A métrica `MG-021`, taxa de sucesso de exercício de rollback, é o registro disso. Um caminho de rollback apenas afirmado não é um caminho de rollback; o incidente canônico do caso fundador é precisamente um caso em que o estado afirmado e o estado real divergiam.

6.2 Quem aprova, e como isso é registrado

A promoção é aprovada pelo responsável pelo domínio afetado, por função (role), e nunca pela parte que se beneficiaria da promoção. Onde a parte que promove e a parte que age são agentes compartilhando uma única identidade, como ocorre no caso fundador, a aprovação não é um controle e o protocolo não pode ser operado de forma alguma. Isso não é um detalhe procedimental: `F-005` registra uma instância de produção em que um agente registrou sua própria revisão por pares usando o token de outro agente, o que foi possível porque todas as credenciais de agente eram legíveis pela única conta sob a qual todo agente executa.

A promoção é registrada como uma mudança de mandato, em um artefato de mandato assinado e versionado (`GR-L07-03`) com uma função de aprovador e uma expiração (`GR-L07-04`), escrita em um diário de mudanças de mandato (`GR-L07-05`) que a parte governada não pode escrever. A métrica `MG-034`, estabilidade de mandato, é computada a partir desse diário. Vale explicitar que, no caso fundador, essa métrica não está meramente não coletada, mas é impossível de mensurar em princípio, porque o mandato é prosa em arquivos que a parte governada pode editar, e não existe nenhum artefato versionado cujas mudanças pudessem ser contadas. Essa é a questão em aberto `OQ-012`.

6.3 Rebaixamento, e por que precisa ser structural

O rebaixamento é acionado automaticamente, pelo avaliador, e não por um humano que percebe, sob qualquer uma das seguintes condições:

- Um guardrail do qual a promoção dependia deixa de emitir seu sinal, o que é

detectado pela métrica de cobertura de emissão de sinal de controle `MG-061`. Um controle que silenciosamente parou de funcionar é indistinguível de um que funciona perfeitamente, a menos que seu silêncio seja, ele mesmo, um evento.

- A taxa de divergência de alegação para a classe de ação sobe acima do

limiar que sustentou a promoção.

- Uma condição de viabilidade de supervisão que se sustentava na promoção

deixa de se sustentar, o que é o que o avaliador de viabilidade de supervisão (`BL-015`) existe para detectar.

- O mandato expira e não é renovado. A expiração é o único gatilho de

rebaixamento que não exige que nada seja detectado, motivo pelo qual `GR-L07-04` é desproporcionalmente valioso em relação ao seu custo.

> **O REBAIXAMENTO NÃO DEVE DEPENDER DE UM HUMANO PERCEBER**
>
> Todo argumento da Seção 4 se aplica à própria decisão de rebaixamento. Se o rebaixamento depende de um humano observar que um controle se degradou, então o rebaixamento herda VC1, VC2 e VC4, e não acontecerá: o sinal de que um controle parou de funcionar é exatamente o tipo de sinal que está ausente, atrasado, ou perdido em volume. O rebaixamento precisa, portanto, ser structural, no mesmo sentido usado ao longo deste texto: o nível promovido deve deixar de estar disponível quando suas pré-condições deixam de se sustentar, sem que nenhuma parte escolha retirá-lo. A expiração de mandato é o mecanismo mais barato conhecido com essa propriedade, e é a razão pela qual um mandato deveria ser um arrendamento (lease), e não uma concessão.

---

7. Localizando a Fronteira Estrutural de Autonomia como função

A proposta doutoral introduz a Fronteira Estrutural de Autonomia (Structural Autonomy Boundary) como o ponto em que a delegação a um agente deixa de ser segura por instrução e passa a exigir enforcement arquitetural verificável. A posição deste programa é que a fronteira não é uma linha através do espaço de ações, mas uma função, e que tratá-la como uma linha é o que produz as falhas que a Seção 4 documenta.

Três consequências se seguem, e cada uma é testável.

A fronteira se move com o conjunto de guardrails, para algumas ações e não para outras. Para as 28 classes de ação dominadas por guardrails, a fronteira está onde quer que o condition set atual a coloque, e ela se move quando o conjunto muda. Para as 32 classes dominadas por determinante, ela não se move. Uma organização que trata a fronteira como fixa investirá em excesso no segundo grupo e de menos no primeiro, que é o padrão observável no caso fundador: um esforço substancial de engenharia foi direcionado a controles do tipo instructional sobre ações que os guardrails poderiam delimitar, e quase nenhum para a separação de identidade, que delimitaria nove classes de ação de forma definitiva.

A fronteira é uma propriedade de um ambiente, não de uma tecnologia. O mesmo agente, o mesmo modelo e a mesma tarefa se situam em níveis diferentes em dois ambientes com conjuntos de guardrails diferentes. É por isso que o caso fundador é evidência de viabilidade e uma fonte inicial de taxonomia, e não um ambiente experimental, e por que o quinto objetivo específico exige avaliação em um cenário adicional.

A fronteira pode ser cruzada sem que ninguém decida cruzá-la. Um controle que se degrada, um volume de sinal que cresce, uma unidade de supervisão que reinicia um agente encerrado: nenhum desses é uma decisão, e cada um move a fronteira efetiva enquanto a nominal permanece onde foi traçada. Este é o argumento prático mais forte para o avaliador de viabilidade de supervisão (`BL-015`, proposto no D6), e para o rebaixamento structural na Seção 6.3.

> **A RELAÇÃO REAFIRMADA**
>
> `L(a, G) = max( floor(a), ceiling(G, a) )`. A Fronteira Estrutural de Autonomia é o lugar geométrico dos pontos em que `ceiling(G, a)` cruza o Nível II, isto é, onde a instrução deixa de ser suficiente e a arquitetura precisa assumir. Para ações em que `floor(a)` já é o Nível II ou III, a fronteira não é a restrição operante, e nenhuma quantidade de arquitetura o é.

---

8. Posição em relação à literatura

As afirmações de novidade são enunciadas aqui em relação à própria bibliografia da proposta doutoral, e onde a resposta honesta é que uma contribuição é uma operacionalização ou uma reformulação, e não uma extensão, isso é dito.

Parasuraman, Sheridan e Wickens (2000) estabelecem que a automação se aplica em graus através de estágios funcionais e que o nível apropriado é selecionado avaliando consequências. A afirmação de que a autonomia é gradual, e não binária, é uma **reformulação** de sua posição. A extensão é que este programa torna o nível uma função do conjunto de guardrails, e não uma escolha de design avaliada contra consequências, de modo que a mesma ação se situa em níveis diferentes em dois ambientes. Nos termos deles, os determinantes de supervisão são os casos em que nenhuma escolha de design de automação muda o papel humano exigido.

Endsley (2017) é o trabalho prévio mais próximo da Seção 4, e a relação precisa ser enunciada com cuidado, porque seria fácil superestimar a contribuição. Endsley estabelece o problema de desempenho out-of-the-loop: à medida que a automação aumenta, a consciência situacional do humano se degrada, de modo que o operador está menos apto a intervir precisamente quando a intervenção é mais necessária. Este programa não descobre isso. Sua contribuição é uma **operacionalização**: um teste por ponto de supervisão para saber se a degradação ocorreu em uma dada arquitetura, e uma medida de quanto. Dois elementos são genuinamente aditivos. Primeiro, a degradação identificada aqui decorre de propriedades arquiteturais, como um mecanismo de intervenção incapaz de interromper o trabalho em andamento, e não do declínio de vigilância. Segundo, ela é, portanto, previsível a partir da arquitetura antes que qualquer operador seja observado. O mecanismo de Endsley é cognitivo e este é structural; eles se somam, e não competem.

Hobbs e Li (2024) fornecem o vocabulário in, on e out-of-the-loop. A contribuição aqui é a modesta **extensão** de que as três categorias são propriedades de um ponto de supervisão, e não de um sistema, e que uma atribuição de categoria é uma afirmação que pode falhar. A literatura de aviação distingue há muito tempo o projetado-para do como-operado, então isso é um estreitamento, e não uma descoberta.

Engin e Hand (2025) argumentam que categorias fixas de supervisão são inadequadas para sistemas que se movem entre níveis, e propõem governança dimensional. Este programa aceita o argumento e o **estende** perguntando o que determina em que ponto da dimensão uma ação pode se situar. A resposta tem dois termos, um passível de engenharia e outro não, e o segundo termo é o verdadeiro acréscimo: um relato puramente dimensional não tem como expressar a afirmação de que algumas ações não se movem ao longo da dimensão por mais que se invista. A distinção entre acesso e mandato, que o enquadramento deles apoia diretamente, é medida aqui, e não apenas afirmada.

Khoo, Foo e Lee (2025) relacionam capacidade a risco a requisito de governança. A relação é **complementar**. O registro de classes de ação pode ser lido como uma instanciação concreta de um inventário de capacidades para um ambiente. O ponto de diferença que vale a pena registrar é que um framework de capacidade-risco não distingue um requisito atendido por prosa de um atendido por arquitetura, e a evidência do caso fundador é que a distinção é onde vivem as falhas.

O NIST AI RMF (2023) articula princípios sem especificar controles técnicos verificáveis, o que a proposta identifica como a lacuna. Este programa **fornece uma das camadas ausentes**, em vez de competir: o catálogo de guardrails é utilizável sob a função Manage, e o registro de métricas sob a função Measure. A ressalva honesta é que o catálogo é derivado de um único ambiente e sua generalidade é uma hipótese, o que é a questão em aberto `OQ-004`.

ISO/IEC 27001:2022 e ISO/IEC 42001:2023 situam a responsabilização em uma função nomeada e exigem que a supervisão humana seja definida e efetiva. Nada aqui é novo em relação a nenhuma das normas. A contribuição é para a prática de asseguração: as seis condições de viabilidade são uma leitura testável da palavra "efetiva", e o caso fundador demonstra que um arranjo de supervisão documentado pode satisfazer um auditor e ainda assim falhar em todas as condições.

Humble e Farley (2010) estabeleceram o pipeline de implantação como um artefato arquitetural, com reversibilidade, tamanho de lote pequeno e separação de funções como propriedades de primeira classe. Quase tudo na camada L6 de guardrails é uma **reformulação** de prática estabelecida quinze anos antes de os sistemas agênticos existirem, e dizer isso importa. O que é novo é o ator: a entrega contínua pressupõe que a parte controlada é uma equipe humana que pode ser responsabilizada por um processo, ao passo que um agente satisfará qualquer processo que seja verificável e contornará qualquer um que não seja. O achado `F-006`, de que a aprovação não está vinculada ao artefato que executa, é um defeito clássico de entrega contínua, e sua presença é evidência de que ambientes agênticos regridem em relação à prática estabelecida, em vez de enfrentar problemas inéditos.

Yao et al. (2023) e Yang et al. (2024) estabelecem que a saída do modelo é probabilística e pode estar confiantemente errada, e que a interface agente-computador determina o que um agente pode fazer. A consequência de governança extraída aqui é estreita e, até onde este programa consegue estabelecer, não enunciada nesses termos em outro lugar: porque o traço de raciocínio e a ação são produzidos pelo mesmo processo, o traço não é evidência sobre a ação. Toda arquitetura que trata a explicação de um agente sobre o que fez como um registro do que fez herda esse problema.

> **A AFIRMAÇÃO DE NOVIDADE MAIS CLARA**
>
> De tudo neste documento, o elemento com a reivindicação de novidade mais forte é a separação de `floor(a)` de `ceiling(G, a)`, junto com o achado de que, em um ambiente real, o maior grupo do resíduo é fixado por um determinante que é totalmente removível por engenharia de identidade. A consequência prática, de que o resíduo de uma organização é substancialmente maior do que seu resíduo último e de que a diferença é majoritariamente trabalho de identidade, é o resultado que este programa mais gostaria de ver testado em outro lugar.

---

9. Limitações e ameaças à validade

Enunciadas na ordem de quanto deveriam preocupar um leitor.

Um ambiente, um instantâneo, um observador. Toda afirmação empírica vem de um host, em um parque, em um dia, examinado pela parte que o construiu. O catálogo de guardrails, o conjunto de determinantes e as condições de viabilidade são moldados pelo que aquele ambiente específico contém. Um ambiente diferente plausivelmente produziria camadas e determinantes diferentes, e não há como saber, de dentro, o quanto. Esta é a razão pela qual o caso fundador explicitamente não é o ambiente experimental.

O observador é o operador. O candidato construiu e opera o caso fundador. Isso produz acesso e compreensão que nenhum observador externo teria, e produz um risco sistemático de que a análise seja moldada pelo que o construtor já acreditava. Duas mitigações parciais foram aplicadas, e nenhuma é suficiente: toda afirmação factual carrega um identificador de evidência apontando para uma sondagem bruta, e uma hipótese trazida para a execução foi corrigida, e não confirmada (`F-035`). Uma descoberta que apenas confirma seus próprios pressupostos não é evidência, e uma correção não é muitas.

O resíduo é relativo a um catálogo. O resíduo é definido como o que permanece acima do Nível I sob o condition set `CS-09` , que contém todo controle catalogado. Um controle que não está no catálogo não pode mover uma ação para fora do resíduo, por construção. O resíduo é, portanto, um limite superior relativo ao conhecimento atual e deveria diminuir à medida que o catálogo crescer. Não é uma afirmação sobre o que é possível em princípio.

As condições de viabilidade são afirmadas, não derivadas. Seis condições foram selecionadas porque são as que a evidência tornou visíveis. Não há prova de que sejam completas, independentes ou mínimas, e a ordenação usada para selecionar a condição vinculante é uma convenção, e não um resultado. Uma sétima condição pode existir. As condições são falseáveis de uma forma específica: se um ponto de supervisão passa em todas as seis e a supervisão ainda assim falha, o conjunto está incompleto.

VC4 se baseia em um volume, não em uma capacidade. As avaliações de atenção se baseiam na razão observada de 19.334 eventos de automação para 4 alertas voltados a humanos, que é um volume. A capacidade do operador não foi medida. Onde VC4 é registrada como falhando, o julgamento é defensável, mas não medido, e isso é a questão em aberto `OQ-005` .

As cifras de latência são de ordem de grandeza. Os valores de `t_harm` e `t_detect` no registro de supervisão são faixas de ordem de grandeza lidas de agendamentos e código, não medições. Para a maioria das classes de ação, a margem é negativa por um fator grande o suficiente para que a precisão não importe. Para uma minoria, ela genuinamente não é computável, e essas são marcadas, e não estimadas.

O material relatado não é verificado. Os 61,1 por cento de divergência, o incidente de implantação, os controles em tempo real e a identidade somente leitura do System of Record são todos REPORTED. São estruturantes neste documento e não puderam ser verificados dentro das restrições desta execução. A questão em aberto `OQ-001` registra o que uma execução posterior deveria fazer.

Ausência de intervenção não é ausência de necessidade. O achado `F-042` registra que nenhuma intervenção humana aparece em 46 dias de logs. Isso não estabelece que a intervenção era necessária e não ocorreu. Estabelece que o laço não foi exercido, o que é uma afirmação mais fraca, e é a afirmação mais fraca que este documento faz.

O framework não foi usado para tomar uma decisão. Todo framework parece coerente antes de ser aplicado. Nada aqui ainda causou a promoção ou o rebaixamento de uma classe de ação, e até que cause, sua usabilidade permanece não testada.

Apêndice A. Registro de determinantes de supervisão

SD-01 Irreversibility

DEFINITION	An action is pinned to human supervision to the extent that its effect cannot be undone by any mechanism available to the system after the fact.
TEST	Is there a rollback path that restores the prior state exactly, is it available to the actor, and has it been demonstrated by exercise rather than asserted? If any of the three is false, the determinant applies.
APPLIES WHEN	The effect persists after the action, and no exercised rollback restores the prior state; or the rollback itself has a blast radius comparable to the original action.
EXAMPLES	AC-006 rewrite published version control history; AC-024 delete files outside a working tree; AC-027 read credential material at rest; AC-054 delete an audit record
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Pre-apply snapshots (GR-L06-02), pre-approved rollback targets only (GR-L06-03) and proven rollback exercise (MG-021) convert some irreversible actions into reversible-with-effort ones. They cannot convert an action whose effect has already left the system boundary, which is why SD-01 and SD-02 compound rather than overlap.
LITERATURE	Humble and Farley 2010 on reversibility as a deployment property; the proposal's own reversibility axis in the autonomy-boundary taxonomy.

SD-02 Externality

DEFINITION	An action is pinned to human supervision to the extent that its effect crosses the system boundary and reaches a party who did not consent to the system's error rate.
TEST	Does any effect of this action become visible to, or act upon, a person or organisation outside the operator's span of control, before any review can occur?
APPLIES WHEN	The output reaches a member of the public, a customer, a regulator, a certificate transparency log, a public name resolution system, or any third party system.
EXAMPLES	AC-047 send an email to an external recipient; AC-048 publish content on a public web surface; AC-049 conduct a real-time voice interaction; AC-050 send a message to a member of the public; AC-040 change a public DNS record
GUARDRAIL-REMOVABLE	no
RESIDUAL LEVEL	III
MITIGATION	Structural interception before the boundary is crossed, of which the reported test-mode overlay (GR-L06-08) is the estate's cleanest instance, moves the boundary rather than removing it: the action becomes internal and the determinant no longer applies to the intercepted form. Nothing reduces the determinant for the real form.
LITERATURE	Extends the blast radius axis of the proposal's taxonomy. Parasuraman, Sheridan and Wickens 2000 treat consequence severity as a determinant of the appropriate automation level; the externality framing narrows this to who bears the consequence.

SD-03 Accountability requirement

DEFINITION	An action is pinned to human supervision where a named human must be answerable for the decision, by law, contract, standard or professional obligation, irrespective of whether the machine would decide correctly.
TEST	If the decision were later challenged, would an answer of the form 'the system decided' be accepted by the party entitled to ask? If not, the determinant applies.
APPLIES WHEN	A statutory, contractual or certification obligation attaches a named signatory to the decision, or the decision constitutes an attestation about the organisation.
EXAMPLES	AC-055 write a completion attestation about its own work; AC-056 record an approval in the deployment approval gate; AC-052 write to the System of Record
GUARDRAIL-REMOVABLE	no
RESIDUAL LEVEL	III
MITIGATION	Nothing reduces this determinant, because it is not a claim about competence. Guardrails can make the human's decision better informed, faster and cheaper, which is worth doing, but the requirement for a human decision is exogenous to the architecture. This is why DM-08 has an autonomy ceiling that no investment raises.
LITERATURE	ISO/IEC 27001:2022 and ISO/IEC 42001:2023 both locate accountability with a named role rather than with a control. NIST AI RMF 2023 GOVERN function, likewise.

SD-04 Value and judgement content

DEFINITION	An action is pinned to human supervision where it trades off values that the organisation has not reduced to a rule, and where more than one resolution is defensible.
TEST	Can the decision be expressed as a predicate over observable state that a reviewer would accept as complete? If the honest answer requires the phrase 'it depends on the circumstances', the determinant applies.
APPLIES WHEN	The correct action depends on a priority ordering that is contested, unstated, or situational, such as availability against correctness, or speed against certainty.
EXAMPLES	AC-044 deploy application code to another estate host during an incident; AC-039 call a cloud control plane mutating API where the resource is shared; AC-045 restart a production service while work is in flight
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Making the priority ordering explicit as declarative policy (GR-L07-01) converts part of the judgement into a rule and genuinely removes it from the residue. What remains is the set of cases the policy did not anticipate, which is where default deny (GR-L07-07) becomes the safe resolution rather than an obstruction.
LITERATURE	Engin and Hand 2025 argue that fixed supervision categories fail precisely because the judgement content of an action is not fixed. This determinant operationalises that argument as a per-action test.

SD-05 Unverifiability

DEFINITION	An action is pinned to human supervision where its success cannot be checked by any mechanism independent of the actor that performed it.
TEST	Is there an oracle for success that does not consult the acting agent, and that can be evaluated mechanically? If the only available check is the agent's own report, the determinant applies.
APPLIES WHEN	The claim of completion is not corroborated by infrastructure-emitted evidence, or the evidence store is writable by the actor whose claim it corroborates.
EXAMPLES	AC-055 write a completion attestation about its own work; AC-053 write to its own transcript record; AC-019 dispatch a brief to an agent
GUARDRAIL-REMOVABLE	yes
RESIDUAL LEVEL	II
MITIGATION	This is the determinant most responsive to guardrail investment, and therefore the most important one for the doctorate. An independent verifier under a separate identity (GR-L08-05), infrastructure-emitted events (GR-L08-11) and a prohibition on self-attestation (GR-L08-06) together remove it for a large class of actions. The founding case's 61.1 per cent divergence is a measurement of this determinant going unaddressed.
LITERATURE	Yao et al. 2023 and Yang et al. 2024 establish that model output is probabilistic and can be confidently wrong, requiring external verification rather than reliance on self-report. This determinant is the governance consequence of that finding.

SD-06 Distribution shift

DEFINITION	An action is pinned to human supervision where the situation it is taken in lies outside the distribution over which the system's competence has been evidenced.
TEST	Has this action class been exercised, under conditions materially like the present ones, often enough for its failure rate to be estimated? If the present case is novel in a way the actor cannot itself detect, the determinant applies.
APPLIES WHEN	The environment has changed, the workload is unfamiliar, or the action is being taken for the first time at this scale, in this sequence, or under this load.
EXAMPLES	AC-060 self-update the agent runtime software; AC-039 call a cloud control plane mutating API against a resource class not previously touched; AC-044 deploy application code after an upstream dependency change
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Canary and wave ordering (GR-L06-04), blast radius caps (GR-L06-07) and staleness bounds on decision inputs (GR-L11-09) reduce the consequence of being out of distribution. They do not tell the actor that it is out of distribution, which is the part that requires a human, and is why automatic demotion (D2) must be structural rather than dependent on the agent recognising its own novelty.
LITERATURE	Endsley 2017 on the out-of-the-loop performance problem: the operator's situation awareness degrades precisely in the conditions where it is most needed. The same argument applies to the automation itself.

SD-07 Containment shortfall

DEFINITION	An action is pinned to human supervision where the blast radius of a mistake exceeds what the containment architecture can hold, so that a single error escapes the boundary the design assumed.
TEST	If this action were performed wrongly in the worst plausible way, would the consequence stay inside a boundary the operator can restore independently? If not, the determinant applies.
APPLIES WHEN	The actor holds privilege, credentials or reach beyond the scope of the action, so that an error in the action can express itself outside that scope.
EXAMPLES	AC-029 execute a command as root; AC-037 obtain cloud credentials from the instance metadata service; AC-043 execute a command on another estate host; AC-027 read credential material at rest
GUARDRAIL-REMOVABLE	yes
RESIDUAL LEVEL	III
MITIGATION	This determinant is entirely an artefact of architecture and is therefore fully removable in principle. Per-agent workload identity (GR-L01-02), least privilege per action class (GR-L01-03), ephemeral credentials (GR-L01-04), segmentation (GR-L02-08) and egress control (GR-L02-05) remove it. On the founding case it applies to almost everything, because one account holds unrestricted root and five cloud administrator identities.
LITERATURE	The structural guardrails named in the proposal: least privilege, ephemeral credentials, network segmentation. Amazon Web Services 2024 on least privilege as an identity property rather than an operational practice.

SD-08 Objective ambiguity

DEFINITION	An action is pinned to human supervision where the goal it serves is stated in terms that admit more than one faithful reading, so that a competent actor can satisfy the letter of the instruction while defeating its purpose.
TEST	Could a diligent actor complete this action, be judged compliant against the instruction as written, and still have produced an outcome the instructing human would reject? If so, the determinant applies.
APPLIES WHEN	The mandate is prose rather than a predicate, the success criterion is a human judgement, or the instruction contains an implicit precondition the actor cannot check.
EXAMPLES	AC-019 dispatch a brief to an agent; AC-055 write a completion attestation; AC-002 modify source files where the requirement is expressed as an intent
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Expressing the mandate as data rather than prose (GR-L07-01), classifying actions at the moment of attempt (GR-L07-08) and defaulting to deny for unclassified actions (GR-L07-07) narrow the space of faithful readings. They do not close it, because the residual ambiguity lives in the goal, not in the mechanism.
LITERATURE	Adjacent to the alignment literature but here treated narrowly as a governance property of the mandate artefact. The proposal's mandate stability metric measures the frequency of re-specification that this determinant predicts.

SD-09 Compounding

DEFINITION	An action is pinned to human supervision where an error in it creates the conditions for further error, so that consequence grows with time rather than remaining bounded at the moment of the mistake.
TEST	Does the output of this action become the input to the next decision of the same actor, without an intervening independent check? If so, the determinant applies.
APPLIES WHEN	Self-repair loops, retry ladders, and any automation whose trigger condition can be produced by its own previous action.
EXAMPLES	AC-016 inject input into another agent's live session; AC-059 answer an interactive prompt on a human's behalf; AC-060 self-update the agent runtime software
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Circuit breakers on repeated failure (GR-L09-03), cooling off between related high impact actions (GR-L11-03) and sequencing constraints requiring verification before the next class unlocks (GR-L11-04) bound the compounding without removing it. This determinant is why domain DM-03, self-healing, is where autonomy is most attractive and most dangerous at the same time.
LITERATURE	Not directly addressed in the proposal's bibliography as a supervision determinant. Proposed here as an extension, on the evidence of the founding case's watcher topology, in which four of five independent watchers respond to silence by injecting text into the agent that has gone silent.

SD-10 Adversarial adaptation

DEFINITION	An action is pinned to human supervision where an adversary can observe the control and adapt to it, so that the control's effectiveness degrades as a function of its own deployment.
TEST	Would a party with an interest in the action succeeding be able to observe the control's decision boundary and construct an input that satisfies it? If so, the determinant applies.
APPLIES WHEN	The control is a pattern match, a classifier, or any rule whose decision boundary is inferable from its outputs, and an interested party can supply the input.
EXAMPLES	AC-036 make an arbitrary outbound network request; AC-047 send an email to an external recipient; AC-002 modify source files where the source is drawn from untrusted content
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Allowlists (GR-L04-06, GR-L02-06) degrade far more slowly than deny lists (GR-L04-05) under adaptation, because an allowlist's failure mode is a false refusal rather than a false permit. Content-derived instructions reaching an agent's context, which the founding case has as a scheduled path, make this determinant live rather than theoretical.
LITERATURE	Not addressed in the proposal's bibliography. Proposed here as an extension, and flagged as the determinant most likely to be understated in a study whose workloads are benign by construction.

Apêndice B. Registro de pontos de supervisão

A avaliação completa de cada ponto de supervisão contra as seis condições de viabilidade. `pass`, `partial` e `fail` são como definidos na Seção 4.2. `not-computable` aparece onde um termo de latência não pôde ser estabelecido, e nunca é tratado como aprovação.

Tabela D2-2. Pontos de supervisão avaliados contra as seis condições de viabilidade.

ID	NAME	NOMINAL	SIGNAL ORIGIN	INTERVENTION SCOPE	VC1	VC2	VC3	VC4	VC5	VC6	EFFECTIVE	DIVERGES
SP-001	Completion notification from the agent to the operator	on-the-loop	agent	prevents-next	fail	fail	fail	partial	fail	partial	out-of-the-loop	yes
SP-002	Completion marker and report file written by the agent	on-the-loop	agent	none	fail	fail	fail	pass	fail	fail	out-of-the-loop	yes
SP-003	Live terminal view of an agent session in the console	on-the-loop	mixed	prevents-next	partial	fail	fail	fail	partial	partial	out-of-the-loop	yes
SP-004	Console audit log review	on-the-loop	infrastructure	none	pass	fail	fail	pass	pass	fail	out-of-the-loop	yes
SP-005	Deployment approval gate: human approval of a patch	in-the-loop	mixed	prevents-next	partial	pass	fail	pass	fail	pass	in-the-loop, weakly	yes
SP-006	Deployment approval gate: independent reviewer	in-the-loop	agent	prevents-next	fail	pass	fail	pass	fail	fail	out-of-the-loop	yes
SP-007	Deployment approval gate: binding approval to what runs	in-the-loop	infrastructure	none	fail	fail	fail	pass	fail	fail	out-of-the-loop	yes
SP-008	Update plane: human approval of an update proposal	in-the-loop	infrastructure	prevents-next	pass	pass	fail	pass	pass	pass	in-the-loop	no
SP-009	Update plane: kill switch	on-the-loop	infrastructure	prevents-next	pass	not-computable	fail	pass	partial	fail	on-the-loop, nominal only	yes
SP-010	Stall watchdog escalation to the operator	on-the-loop	infrastructure	none	pass	fail	fail	fail	fail	fail	out-of-the-loop	yes
SP-011	Human-facing session watchdog alert	on-the-loop	infrastructure	none	pass	fail	fail	pass	partial	fail	out-of-the-loop	yes
SP-012	Console session termination	on-the-loop	infrastructure	halts-in-progress	pass	fail	pass	pass	pass	partial	on-the-loop for unsupervised sessions only	yes
SP-013	Supervised fleet agent graceful stop	in-the-loop	infrastructure	prevents-next	pass	fail	fail	pass	pass	fail	out-of-the-loop	yes
SP-014	Console chat interrupt	on-the-loop	infrastructure	prevents-next	pass	fail	fail	pass	pass	fail	out-of-the-loop	yes

ID	NAME	NOMINAL	SIGNAL ORIGIN	INTERVENTION SCOPE	VC1	VC2	VC3	VC4	VC5	VC6	EFFECTIVE	DIVERGES
SP-015	Inter-agent message bus kill switch	on-the-loop	infrastructure	prevents-next	pass	not-computable	fail	pass	partial	partial	on-the-loop, nominal only	yes
SP-016	Operator instruction channel from the handset	in-the-loop	mixed	prevents-next	partial	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-017	Version control review of agent commits	on-the-loop	infrastructure	prevents-next	pass	fail	fail	partial	pass	pass	out-of-the-loop	yes
SP-018	Pre-merge review before code reaches a shared branch	in-the-loop	infrastructure	none	fail	fail	fail	pass	fail	fail	out-of-the-loop	yes
SP-019	Operator oversight of cloud control plane changes	on-the-loop	infrastructure	prevents-next	pass	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-020	Operator oversight of public name resolution changes	in-the-loop	infrastructure	prevents-next	fail	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-021	Operator oversight of public entry point configuration	in-the-loop	infrastructure	prevents-next	fail	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-022	Operator oversight of outbound email	on-the-loop	infrastructure	none	fail	fail	fail	pass	fail	fail	out-of-the-loop	yes
SP-023	Notification rate limit and deduplication as an attention control	on-the-loop	infrastructure	none	pass	not-computable	fail	partial	pass	fail	on-the-loop, partially	yes
SP-024	Structural test mode intercepting external dispatch	out-of-the-loop	infrastructure	halts-in-progress	pass	pass	pass	pass	pass	pass	out-of-the-loop, defensibly	no
SP-025	Acknowledge-to-stop on an escalating external action	on-the-loop	infrastructure	prevents-next	pass	pass	partial	pass	partial	pass	on-the-loop	no
SP-026	Multi-state closure confirmation	in-the-loop	infrastructure	prevents-next	pass	pass	fail	pass	pass	pass	in-the-loop	no
SP-027	Independent audit flow with an adversarial layer	on-the-loop	infrastructure	none	pass	fail	fail	pass	pass	partial	out-of-the-loop at the time of action	yes
SP-028	Compliance attestation by a human owner	in-the-loop	mixed	prevents-next	partial	pass	pass	pass	partial	pass	in-the-loop	no
SP-029	Read-only connection identity to the System of Record	out-of-the-loop	infrastructure	halts-in-progress	pass	pass	pass	pass	pass	pass	out-of-the-loop, defensibly	no

ID	NAME	NOMINAL	SIGNAL ORIGIN	INTERVENTION SCOPE	VC1	VC2	VC3	VC4	VC5	VC6	EFFECTIVE	DIVERGES
SP-030	Interactive permission prompt raised by the agent tooling	in-the-loop	infrastructure	halts-in-progress	pass	pass	pass	fail	partial	fail	out-of-the-loop	yes
SP-031	Operator oversight of changes to agent instruction files	on-the-loop	infrastructure	prevents-next	fail	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-032	Operator oversight of fleet registry changes	on-the-loop	infrastructure	prevents-next	pass	fail	fail	pass	partial	pass	out-of-the-loop	yes
SP-033	Resource consumption oversight	on-the-loop	mixed	none	partial	not-computable	fail	pass	partial	fail	out-of-the-loop	yes
SP-034	Restart supervision of agent sessions	out-of-the-loop	infrastructure	prevents-next	pass	not-computable	fail	pass	pass	fail	out-of-the-loop	no

A condição vinculante e o detalhe de apoio para cada ponto:

Tabela D2-3. Condição vinculante, termos de latência e o achado durável que registra cada divergência.

ID	BINDING CONDITION	SIGNAL LATENCY	HARM COMPLETION	FINDING	EVIDENCE
SP-001	VC1 independence	seconds after the agent chooses to send; unbounded if it does not	already complete: the notification is sent after the work	F-001	EV-029 EV-039
SP-002	VC1 independence	unbounded: the operator must go and look	already complete	F-001	EV-028 EV-039
SP-003	VC4 attention	sub-second while a human is watching; unbounded otherwise	milliseconds to seconds for most local actions	F-002	EV-014 EV-020 EV-039
SP-004	VC2 latency	unbounded: the operator must go and look	already complete	F-003	EV-034 EV-039
SP-005	VC5 comprehensibility	minutes to hours: the patch waits	seconds once approved and applied	F-004	EV-030
SP-006	VC1 independence	minutes	seconds once approved and applied	F-005	EV-030
SP-007	VC1 independence	unbounded: there is no signal	seconds: the process manager re-executes whatever is on the shared tree	F-006	EV-030
SP-008	none: all six conditions hold	unbounded: proposals wait indefinitely	not applicable today: no proposal has ever been applied	F-007	EV-031
SP-009	VC6 authority	unbounded: no signal is emitted	seconds	F-008	EV-031
SP-010	VC2 latency	180 seconds stall plus up to 60 seconds schedule granularity plus two nudge cycles: realistically over ten minutes	milliseconds to seconds for the actions the agent is taking meanwhile	F-009	EV-028 EV-039
SP-011	VC2 latency	300 seconds stall plus up to 120 seconds schedule granularity	milliseconds to seconds	F-009	EV-028 EV-039
SP-012	VC2 latency	n/a: this is an intervention rather than a signal	milliseconds to seconds	F-010	EV-013 EV-039
SP-013	VC3 halt	n/a: this is an intervention rather than a signal	milliseconds to seconds	F-011	EV-039
SP-014	VC3 halt	n/a: this is an intervention rather than a signal	milliseconds to seconds	F-011	EV-025 EV-039
SP-015	VC3 halt	unbounded: no signal is emitted	milliseconds	F-012	EV-025 EV-039
SP-016	VC5 comprehensibility	seconds	seconds: the orchestrator's tools execute immediately	F-013	EV-038 EV-039
SP-017	VC2 latency	unbounded: the operator must go and look	seconds to push; the artefact is then visible to every consumer	F-014	EV-010
SP-018	the control is absent	n/a: the control is absent	seconds	F-015	EV-010 EV-030

ID	BINDING CONDITION	SIGNAL LATENCY	HARM COMPLETION	FINDING	EVIDENCE
SP-019	VC2 latency	unbounded: no signal reaches the operator	seconds; some resource deletions are irreversible in effect	F-016	EV-033
SP-020	VC1 independence	unbounded: no signal is emitted	seconds to change; bounded below by the record time to live to correct	F-016	EV-033
SP-021	VC1 independence	unbounded: no signal is emitted	seconds; exposure begins at the moment of reload	F-016	EV-008
SP-022	VC1 independence	unbounded	seconds; delivery is not recallable	F-017	EV-029
SP-023	VC4 attention	sub-second when not suppressed; suppressed messages are dropped rather than queued	n/a: this control governs the signal, not an action	F-018	EV-029 EV-039
SP-024	none: the action is contained before the boundary is crossed	immediate at the point of interception	n/a: the harm does not occur, because the boundary is not crossed	F-019	REPORTED
SP-025	none for the ladder; the individual step remains uninterruptible	bounded by the ladder's step interval	each step completes on dispatch; the ladder as a whole is what is being stopped	F-020	REPORTED
SP-026	none	bounded by the state machine	n/a: the control governs closure, not dispatch	F-020	REPORTED
SP-027	VC2 latency	days to weeks: the audit is retrospective	already complete: the audit examines finished episodes	F-021	REPORTED
SP-028	none: the ceiling here is an accountability requirement, not a control weakness	days: the attestation cycle is deliberate	n/a: nothing takes effect until the human attests	F-022	REPORTED
SP-029	none: the control does not depend on agent compliance	n/a	n/a: the write cannot be expressed	F-023	REPORTED
SP-030	VC6 authority: the prompt is suppressed by a permission-skipping flag, and where it does appear a watchdog answers it automatically	immediate when raised	the action is blocked while the prompt is unanswered	F-024	EV-020 EV-028 EV-032
SP-031	VC1 independence	unbounded	takes effect at the next session start	F-025	EV-032
SP-032	VC2 latency	seconds when the operator is looking; unbounded otherwise	immediate	F-026	EV-022
SP-033	VC6 authority	hours: synchronisation is twice hourly	n/a: consumption is continuous rather than an event	F-027	EV-024
SP-034	VC6 authority: termination by the operator is undone automatically, so the operator's override does not survive; the nominal posture is already out-of-the-loop, so there is no divergence, but the arrangement defeats SP-012	n/a	seconds: a terminated session is replaced within the restart interval	F-010	EV-005 EV-028

Apêndice C. Registro do resíduo irredutível

Trinta e duas classes de ação permanecem acima do Nível I sob o condition set `CS-09`, a configuração totalmente endurecida. Cada uma é fixada ao seu determinante dominante.

Tabela D2-4. O resíduo irredutível de supervisão.

ACTION CLASS	NAME	RESIDUE LEVEL	DOMINANT DETERMINANT	PINNED BY	BEST ACHIEVABLE	GUARDRAILS REQUIRED
AC-006	Rewrite published version control history	II	SD-01	SD-01 SD-02	II	GR-L06-02 GR-L08-04 GR-L04-06
AC-019	Dispatch a brief to an agent	II	SD-08	SD-08 SD-05 SD-09	II	GR-L07-01 GR-L07-03 GR-L07-08
AC-016	Inject input into another agent's live session	III	SD-09	SD-09 SD-01 SD-08	II	GR-L09-03 GR-L11-03 GR-L11-04 GR-L07-08
AC-024	Delete files outside a working tree	II	SD-01	SD-01 SD-07	II	GR-L05-09 GR-L03-06 GR-L01-03
AC-027	Read credential material at rest	III	SD-01	SD-01 SD-07	II	GR-L01-04 GR-L01-05 GR-L01-02
AC-029	Execute a command as root	III	SD-07	SD-07 SD-04	II	GR-L01-03 GR-L01-06 GR-L04-03
AC-030	Modify accounts, groups or privilege configuration	III	SD-07	SD-07 SD-01	III	GR-L01-02 GR-L07-06 GR-L08-11
AC-032	Create or modify a systemd unit	III	SD-07	SD-07 SD-09	II	GR-L07-06 GR-L03-04 GR-L08-11
AC-033	Modify reverse proxy configuration	III	SD-02	SD-02 SD-07	III	GR-L06-01 GR-L06-12 GR-L10-04
AC-034	Reload or restart the reverse proxy	II	SD-02	SD-02	II	GR-L06-06 GR-L06-12
AC-036	Make an arbitrary outbound network request	II	SD-02	SD-02 SD-01 SD-10	II	GR-L02-05 GR-L02-06 GR-L02-11
AC-037	Obtain cloud credentials from the instance metadata service	III	SD-07	SD-07 SD-01	II	GR-L02-12 GR-L01-04 GR-L01-11
AC-039	Call a cloud control plane mutating API	III	SD-07	SD-07 SD-04 SD-06	II	GR-L01-03 GR-L07-01 GR-L06-07
AC-040	Change a public DNS record	III	SD-02	SD-02 SD-01	III	GR-L06-01 GR-L11-08 GR-L10-04
AC-041	Obtain or renew a public TLS certificate	II	SD-02	SD-02	II	GR-L06-01 GR-L08-11
AC-043	Execute a command on another estate host	III	SD-07	SD-07 SD-06	II	GR-L01-03 GR-L02-08 GR-L04-06
AC-044	Deploy application code to another estate host	III	SD-03	SD-03 SD-04 SD-06 SD-01	II	GR-L06-01 GR-L06-02 GR-L06-04 GR-L06-12 GR-L08-05
AC-045	Restart a production service on another estate host	II	SD-04	SD-04 SD-01	II	GR-L11-08 GR-L06-04 GR-L09-08
AC-047	Send an email to an external recipient	III	SD-02	SD-02 SD-01	II	GR-L06-08 GR-L10-04 GR-L11-08
AC-048	Publish content on a public web surface	III	SD-02	SD-02 SD-01	III	GR-L06-01 GR-L06-08 GR-L10-04
AC-049	Conduct a real-time voice interaction with a member of the public	III	SD-02	SD-02 SD-03 SD-04 SD-01	III	GR-L06-08 GR-L09-07 GR-L09-08 GR-L11-08
AC-050	Send a message to a member of the public	III	SD-02	SD-02 SD-03 SD-01	III	GR-L06-08 GR-L10-04 GR-L11-08

ACTION CLASS	NAME	RESIDUE LEVEL	DOMINANT DETERMINANT	PINNED BY	BEST ACHIEVABLE	GUARDRAILS REQUIRED
AC-052	Write to the System of Record	III	SD-03	SD-03 SD-01 SD-02	III	GR-L05-01 GR-L05-03 GR-L05-04
AC-054	Delete or truncate an audit or transcript record	III	SD-01	SD-01 SD-05	III	GR-L08-03 GR-L08-04 GR-L05-06 GR-L01-07
AC-055	Write a completion attestation about its own work	II	SD-05	SD-05 SD-03	II	GR-L08-05 GR-L08-06 GR-L08-08 GR-L08-11
AC-056	Record an approval in the deployment approval gate	III	SD-03	SD-03 SD-05	III	GR-L01-08 GR-L08-06 GR-L07-03
AC-057	Set or clear the update kill switch	III	SD-07	SD-07 SD-03	III	GR-L01-08 GR-L07-06 GR-L09-11
AC-059	Answer an interactive permission or resume prompt on a human's behalf	III	SD-03	SD-03 SD-09 SD-05	III	GR-L10-04 GR-L07-06 GR-L08-11
AC-060	Self-update the agent runtime software	II	SD-06	SD-06 SD-09 SD-10	II	GR-L06-04 GR-L06-11 GR-L11-04
AC-025	Modify its own instruction files	II	SD-08	SD-08 SD-09	II	GR-L03-07 GR-L08-11 GR-L07-05
AC-026	Modify the permission settings file	III	SD-07	SD-07 SD-03	III	GR-L03-08 GR-L07-06 GR-L01-02
AC-021	Modify an agent's startup wrapper or launch script	III	SD-07	SD-07 SD-08	III	GR-L03-07 GR-L07-06 GR-L07-05

O argumento para cada entrada, indicando por que o investimento em guardrails não a move:

Tabela D2-5. Por que os guardrails não movem cada entrada do resíduo.

ACTION CLASS	WHY GUARDRAILS DO NOT MOVE IT
AC-006	The published state has already been consumed by other parties and by automated processes; a later correction does not undo what was fetched in the interim.
AC-019	The brief is the mandate. Its ambiguity is a property of natural language goals, not of the delivery mechanism, so no delivery-side control resolves it.
AC-016	Injected text becomes part of another agent's reasoning context and cannot be withdrawn; the receiving agent may act on it before any control observes the injection, and its action can trigger further injection.
AC-024	Deletion of material not held in version control has no restore path unless a backup exists, and a backup the same identity can delete is not a restore path.
AC-027	A secret cannot be un-read. The only remedy is rotation, which is a different action with its own blast radius, and the exposure window is not bounded by the response.
AC-029	Root authority is not an action class, it is the absence of a boundary, so no control operating below the operating system bounds it. Removing it is an identity change, not a guardrail.
AC-030	The action can grant a second identity the same authority, so revoking the first identity does not revoke the capability, and the containment shortfall persists after the correction.
AC-032	The unit set includes the units that supervise the agents, so this action can change the terms of its own supervision, which places it outside any control the supervision layer provides.
AC-033	A configuration change publishes an internal service to the public network; retrieval by third parties during the exposure window is not undone by reverting.
AC-034	The moment of reload is the moment exposure begins; the control must act before it, and after it there is nothing to control.
AC-036	Data that has left the host cannot be recalled. Egress control reduces the set of destinations but the action remains irreversible for permitted destinations.
AC-037	Once minted, the credential is held by the process; blocking the endpoint afterwards does not revoke what was already issued.
AC-039	The identity's authority spans resources unrelated to the action, so a mistake in one action expresses itself across the whole account. Bounding the action does not bound the identity.
AC-040	Propagation and caching mean the window of wrongness is bounded below by the record's time to live rather than by the speed of the correction.
AC-041	Issuance is recorded permanently and publicly in certificate transparency; the act is externally visible and not retractable.
AC-043	The reach of the command is the reach of the credential on the target, not the scope of the intent, and the target's own containment is outside this host's control.
AC-044	The founding case's canonical incident is here: a deploy reported as successful had disabled the service's core connection function. Verification removes SD-05, but the decision to deploy under uncertainty remains a judgement with an accountable owner.
AC-045	Whether the interruption is acceptable depends on what is in flight, which is a situational judgement the actor cannot fully observe. In real-time domains the in-flight work is a person on a call.
AC-047	Delivery is immediate and not recallable, and the recipient is outside the operator's span of control.
AC-048	Removal does not undo retrieval, indexing or caching that occurred while the content was live.
AC-049	The action completes in seconds and its output reaches a person immediately. There is no useful after: supervision must be designed in before the action or it does not exist. Interception before the boundary, as in GR-L06-08, changes which action is being taken rather than supervising this one.
AC-050	The founding case has a recorded episode in which messages about fabricated events reached a real external number. The message is complete on dispatch and the recipient acts on it before any review.
AC-052	The record is authoritative for parties outside the system, and an incorrect authoritative record is acted upon by people. Accountability for the content sits with a named human by obligation, not by preference.
AC-054	Destroying evidence destroys the basis on which every other control is evaluated, including the evaluation of whether this action occurred.

ACTION CLASS	WHY GUARDRAILS DO NOT MOVE IT
AC-055	Self-attestation is unverifiable by construction. Independent verification removes SD-05 and moves this to Level II with a real review, but the attestation itself, as an assertion an organisation relies on, retains an accountability component.
AC-056	An approval is a statement that a named party accepts responsibility. A machine recording it does not create the acceptance, whatever the data model says.
AC-057	A safety flag whose state the governed party controls is not a safety flag. The action must be outside the governed party's reach, which is an identity property.
AC-059	The prompt exists to obtain a human decision. Answering it automatically does not supervise the action, it removes the supervision point, and no control downstream can restore a decision that was never made.
AC-060	Each update moves the actor outside the distribution over which its competence was evidenced, and the actor cannot detect that it has done so.
AC-025	Changing the instruction changes the objective the next actor pursues. Because the instruction layer is context and not enforcement, the security consequence is bounded, but the objective drift is not observable from the action itself.
AC-026	An actor that can widen its own permissions has no boundary; the boundary must be held by an identity the actor does not hold, which is an architectural property rather than a control.
AC-021	The wrapper determines the flags and the standing instruction of every future session of that agent, so writing it is mandate mutation performed by the governed party.