UNIVERSIDADE FEDERAL FLUMINENSE

JESSICA LOUISE MONÇÔRES DE ALMEIDA

Governança de Dados em Sistemas-de-Sistemas por meio de Dados de Proveniência

NITERÓI

2025

JESSICA LOUISE MONÇÔRES DE ALMEIDA

Governança de Dados em Sistemas-de-Sistemas por meio de Dados de Proveniência

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação

Orientador:
Daniel Cardoso Moraes de Oliveira

Coorientador:
Vanessa Braganholo Murta

NITERÓI

2025

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

A447g Almeida, Jessica Louise Monçôres de
Governança de dados em sistemas-de-sistemas por meio de dados de proveniência / Jessica Louise Monçôres de Almeida. - 2025.
57 f.: il.

Orientador: Daniel Cardoso Moraes de Oliveira.
Coorientador: Vanessa Braganholo Murta.
Dissertação (mestrado)-Universidade Federal Fluminense, Instituto de Computação, Niterói, 2025.

1. Dados de proveniência. 2. Governança de dados. 3. Banco de dados de grafos. 4. Sistemas de sistemas. 5. Produção intelectual. I. Oliveira, Daniel Cardoso Moraes de, orientador. II. Murta, Vanessa Braganholo, coorientadora. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

JESSICA LOUISE MONÇÔRES DE ALMEIDA

Governança de Dados em Sistemas-de-Sistemas por meio de Dados de Proveniência

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação.

Aprovada em Outubro de 2025.

BANCA EXAMINADORA

Prof. Daniel Cardoso Moraes de Oliveira, D.Sc. - Orientador, UFF

Prof. Vanessa Braganholo Murta, D.Sc., - Coorientadora, UFF

Prof. Marcos Vinicius Naves Bêdo, D.Sc., UFF

Prof. Kelly Rosa Braghetto, D.Sc., IME/USP

Prof. Victor Ströele de Andrade Menezes, D.Sc., UFJF

Niterói

2025

Vida longa e próspera! (Spock) "Homenagem a Leonard Nimoy"

Agradecimentos

Ao meu orientador e minha orientadora, que me mostraram os caminhos a serem seguidos e pela confiança depositada.

Agradeço à Marinha do Brasil pela oportunidade de aprofundar meus conhecimentos em minha área de formação, por meio desta importante capacitação.

Por fim, agradeço a todos que contribuíram para o êxito desta dissertação.

Resumo

O crescimento do uso de Sistemas-de-Sistemas (SoS), caracterizados pela integração de sistemas autônomos que cooperam para alcançar capacidades superiores, tem gerado novos desafios na governança de dados, especialmente em contextos complexos e distribuídos. A principal dificuldade reside na rastreabilidade, qualidade e integridade dos dados. Uma vez que informações geradas por um sistema são frequentemente reutilizadas por outros, a ausência de mecanismos eficazes de controle compromete a confiabilidade e a auditoria do ciclo de vida dos dados. Nesse contexto, esta dissertação propõe a abordagem `PROVGov-SoS` como uma solução baseada na gerência de dados de proveniência, com o objetivo de estruturar o fluxo informacional entre os sistemas de modo a tornar visível a origem e as transformações dos dados, por meio da captura, persistência e consulta a dados de proveniência modelados conforme o padrão W3C PROV. Ao permitir que usuários e agentes compreendam o ciclo de vida da informação, a `PROVGov-SoS` fortalece a transparência, a responsabilização e a segurança dos dados em ambientes interconectados. A abordagem foi avaliada por meio de um estudo de viabilidade em um SoS real e os resultados demonstraram sua capacidade de oferecer visões explicativas sobre o fluxo de dados, apoiar a análise da trajetória das informações e facilitar a identificação de inconsistências.

Palavras-chave: proveniência, governança de dados, banco de dados de grafos, sistemas-de-sistemas.

Abstract

The growing use of Systems of Systems (SoS), characterized by the integration of autonomous systems that cooperate to achieve superior capabilities, has created new challenges in data governance, especially in complex and distributed contexts. The main difficulty lies in data traceability, quality, and integrity. Since information generated by one system is often reused by others, the absence of effective control mechanisms compromises the reliability and auditability of the data life cycle. In this context, this dissertation proposes the *PROVGov-SoS* approach as a solution based on provenance data management, with the goal of structuring the information flow among systems so as to make the origin and transformations of data visible through the capture, persistence, and querying of provenance data modeled according to the W3C PROV standard. By enabling users and agents to understand the information life cycle, *PROVGov-SoS* strengthens transparency, accountability, and data security in interconnected environments. The approach was evaluated through a feasibility study in a real SoS, and the results demonstrated its ability to provide explanatory views of data flows, support the analysis of information trajectories, and facilitate the identification of inconsistencies.

Keywords: provenance, data governance, graph databases, system of systems.

Lista de Figuras

Lista de Tabelas

Sumário

1 Introdução

Na última década, tem-se observado um aumento no desenvolvimento dos chamados Sistemas de Sistemas (SoS) (Maier, 1998; Cavalcante; Batista; Oquendo, 2024). Os SoSs consistem na integração de sistemas de informação autônomos, cuja interoperabilidade é viabilizada por meio da definição de fluxos de dados (*i.e.*, *dataflows*) entre os sistemas envolvidos. Os SoSs representam uma evolução arquitetural em relação aos sistemas *stand-alone* existentes ao promoverem a interconexão entre sistemas inicialmente independentes (Cavalcante; Batista; Oquendo, 2024). Nesse contexto, um SoS é caracterizado por sua natureza colaborativa onde cada sistema componente mantém sua capacidade de operar de forma independente, mas podendo também atuar junto com os demais sistemas para alcançar objetivos globais que não seriam viáveis de forma isolada. Além disso, os SoSs são, em sua maioria, caracterizados pela complexidade e distribuição geográfica. Uma característica essencial de um SoS é a capacidade de seus componentes serem adicionados ou removidos sem comprometer os comportamentos emergentes desejados no sistema como um todo, *i.e.*, o princípio de independência funcional e interoperabilidade entre os sistemas constituintes (Maier, 1998).

Embora os SoSs ofereçam vantagens como o reúso de componentes, a resiliência arquitetural e a capacidade de integrar múltiplas tecnologias heterogêneas, eles enfrentam desafios no que se refere à governança de dados (Curry; Sheth, 2018). Em um SoS, os dados podem ser recebidos e processados a partir de múltiplos sistemas que compõem o SoS, criando um ecossistema complexo de dados, em que a origem do dado pode ser tanto intraorganizacional quanto interorganizacional (Curry; Scerri; Tuikka, 2022; Curry; Sheth, 2018; Lis; Otto, 2020). Entretanto, a literatura aponta a necessidade de mecanismos para o controle do ciclo de vida dos dados dentro dos SoS (Curry; Sheth, 2018). Esse controle é fundamental, especialmente no contexto de auditoria e conformidade, em que se exige a rastreabilidade dos dados manipulados ao longo do tempo. Diferentemente dos sistemas *stand-alone*, nos quais os dados são centralizados e acessíveis de forma contínua, em um SoS os dados encontram-se distribuídos entre os diversos sistemas participantes.

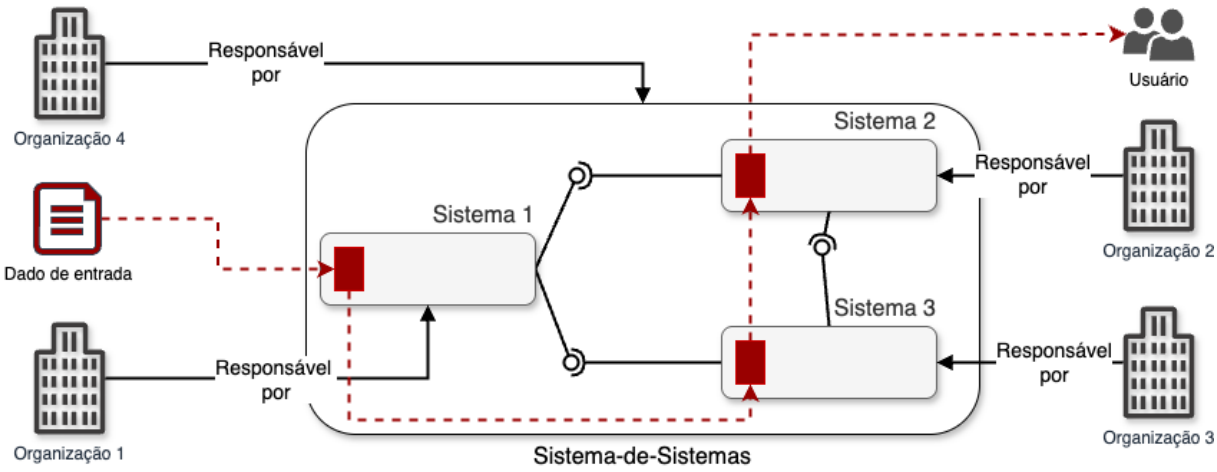


Figura 1: Rastreabilidade dos dados dentro de um SoS.

Isso implica que a auditoria de determinado dado, sem a existência de um caminho claro de derivação e transformação, torna-se praticamente inviável. A ausência de visibilidade sobre esse caminho de derivação dos dados compromete não apenas a confiabilidade dos dados, mas também a governança global do SoS. A Figura 1 apresenta um SoS composto de três sistemas independentes onde o dado consumido pelo usuário seguiu um fluxo de transformação pelo Sistema 1, Sistema 3 e Sistema 2, até que fosse entregue ao usuário.

A literatura já apresenta soluções voltadas ao monitoramento em SoSs. Vierhauser et al. (2016) propõem o *framework* REMINDS, que oferece um modelo de monitoramento em tempo real do estado de um SoS por meio do uso de uma linguagem específica de domínio. Em outra abordagem, Kong et al. (2020) introduzem um método de monitoramento em tempo de execução baseado na extração de traços de execução de sistemas de *software*, utilizando sensores para capturar eventos como chamadas de sistema, interrupções e trocas de contexto, com o objetivo de reduzir ao mínimo possível a interferência sobre os sistemas monitorados. Já Calabro et al. (2021) exploram a integração de múltiplas soluções de monitoramento em tempo de execução aplicadas a diferentes domínios, como o monitoramento de tráfego urbano e sistemas voltados ao setor de saúde. Embora essas abordagens representem avanços, elas se concentram em aspectos específicos, como a análise de eventos de execução e o monitoramento do estado operacional dos sistemas. No entanto, tais abordagem não focam, de forma explícita, em questões relacionadas ao monitoramento das transformações de dados, à rastreabilidade das informações ao longo do SoS e à governança de dados distribuídos. Esses aspectos, fundamentais para a confiabilidade e auditabilidade dos dados no SoS, permanecem como lacunas.

Os dados de proveniência (Herschel; Diestelkämper; Ben Lahmar, 2017) se mostram

como uma solução natural para representar o caminho de derivação dos dados em SoSs (Gammack; Scott; Chapman, 2016; Allen *et al.*, 2011). O uso dos dados de proveniência nesse contexto se encontra em consonância com um dos princípios associados aos seu uso: tornar os sistemas responsabilizáveis por suas ações e fornecer subsídios aos usuários para a avaliação da confiabilidade dos dados produzidos e consumidos (Moreau; Batlajery *et al.*, 2017). Dessa forma, os dados de proveniência podem desempenhar um papel central para uma governança de dados no contexto de SoSs. Se estruturados adequadamente e seguindo os padrões internacionalmente reconhecidos, esses metadados permitem que os dados sejam rastreados ao longo de todo o seu ciclo de vida. Isso garante não apenas a transparência necessária para promover a confiança entre os sistemas componentes, mas também o atendimento a requisitos regulatórios e de negócios, especialmente em domínios críticos e sensíveis (Fu *et al.*, 2011; Gammack; Scott; Chapman, 2016).

Com o objetivo de suprir as lacunas identificadas anteriormente, este artigo propõe a `PROVGov-SoS`, uma abordagem voltada à governança de dados em SoSs por meio da captura, persistência e consulta a dados de proveniência. A `PROVGov-SoS` tem como premissa o uso de dados de proveniência como fio condutor para o monitoramento em grão fino do fluxo de informações entre os diversos sistemas componentes. Para isso, a abordagem realiza o monitoramento de *logs* e comportamentos dos sistemas constituintes do SoS para interceptação e extração de eventos de transformação de dados de interesse. A modelagem e visualização dos dados de proveniência seguem a recomendação W3C PROV (Groth; Moreaun, 2013), garantindo conformidade com padrões e facilitando a interoperabilidade com outras ferramentas compatíveis com o PROV. A persistência dos dados é realizada em um banco de dados orientado a grafos, o que possibilita a execução de consultas em tempo real e *post-mortem*.

O principal objetivo da `PROVGov-SoS` é oferecer uma visão global, transparente e em grão fino do caminho de derivação dos dados no SoS, possibilitando a rastreabilidade e a visualização das transformações aplicadas aos dados ao longo do tempo. No contexto da governança de dados, essa capacidade permite que administradores e demais partes interessadas compreendam de maneira estruturada e auditável o ciclo de vida dos dados, promovendo maior confiança, conformidade regulatória e suporte à tomada de decisão. A `PROVGov-SoS` foi avaliada por meio de um estudo de viabilidade conduzido em um SoS real, composto por sistemas acadêmicos e administrativos de uma universidade pública brasileira, bem como por sistemas externos, como a plataforma Lattes. Esse SoS consolida dados oriundos de diferentes fontes institucionais e governamentais. A avaliação consistiu na captura de dados de proveniência ao longo dos processos de importação, transformação

e integração dos dados provenientes das diversas fontes. Todos os metadados gerados foram estruturados e armazenados em um banco de dados orientado a grafos, permitindo a submissão de consultas analíticas. Uma série de consultas foi executada sobre o banco de dados de proveniência com base em problemas reais reportados por usuários do sistema. As consultas buscaram responder questões relacionadas à origem dos dados, etapas de transformação e integridade das informações. Os resultados evidenciaram a capacidade da `PROVGov-SoS` em oferecer visões explicativas sobre o fluxo de dados, apoiar a análise da trajetória das informações e facilitar a resolução de dúvidas e inconsistências percebidas pelos usuários.

Esta dissertação está estruturada em sete capítulos além desta Introdução. O Capítulo 2 fornece detalhes sobre o contexto teórico necessário ao entendimento da abordagem proposta nesse trabalho. O Capítulo 3.1 revisa trabalhos relacionados. O Capítulo 4 detalha a abordagem do `PROVGov-SoS`, enquanto o Capitulo 5 apresenta uma avaliação do uso dessa abordagem em um SoS real. Por fim, o Capítulo 6 conclui esta dissertação, sintetizando os resultados obtidos e discutindo potenciais trabalhos futuros.

2 Referencial Teórico

2.1 Sistemas-de-Sistemas (SoS)

Os SoSs são conjuntos de entidades autônomas que interagem para alcançar um propósito comum. Embora operem independentemente em suas próprias capacidades e sob seus próprios mecanismos de gerenciamento, esses sistemas colaboram para constituir uma entidade maior e mais complexa. Eles estão presentes em diversos setores, incluindo transporte (Lee *et al.*, 2019), redes elétricas (Uslar *et al.*, 2019; Ibne Hossain *et al.*, 2020), operações militares (George; Santhanakrishnan *et al.*, 2019; Wu; Wu; Sun, 2021) e políticas públicas (Ammara *et al.*, 2022). Eles apresentam um conjunto único de desafios que os distinguem de sistemas tradicionais. Estes desafios surgem predominantemente da independência operacional e gerencial de seus sistemas constituintes, sua natureza evolutiva e os comportamentos emergentes que exibem. Diversas características e desafios têm sido identificados nesse contexto.

Um dos aspectos fundamentais é a interação entre autonomia e colaboração. Cada sistema constituinte dentro de um SoS preserva sua independência operacional. Contudo, para alcançar objetivos compartilhados, esses sistemas atuam em sinergia, estabelecendo um equilíbrio essencial entre a independência individual e a cooperação mútua. Adicionalmente, a heterogeneidade e a necessidade de interoperabilidade representam desafios cruciais. Os componentes de um SoS podem vir de várias fontes e plataformas de hardware/software, ter sido desenvolvidos em linguagens distintas e projetados sob metodologias diferentes. Garantir a interoperabilidade entre esses elementos heterogêneos em múltiplos níveis (técnico, social, organizacional e legal) é um desafio para o projeto, construção e evolução de um SoS (Cavalcante; Batista; Oquendo, 2024; Tekinerdogan, 2022).

A complexidade é um desafio significativo, especialmente no que tange ao comportamento emergente. A interação entre múltiplos sistemas independentes e frequentemente heterogêneos resulta em um comportamento no nível do SoS que não se manifesta em ne-

nhum dos sistemas componentes individualmente (Cavalcante; Batista; Oquendo, 2024; Sage; Cuppan, 2001). Essas interações podem gerar comportamentos emergentes, cuja previsão é difícil pela análise isolada dos componentes. Embora esse comportamento emergente seja, muitas vezes, o propósito da criação de um SoS, sua previsão, compreensão e gerenciamento podem ser problemáticos (Tekinerdogan, 2022). Lidar com esses comportamentos, particularmente os indesejados ou inesperados, constitui um desafio notável, pois o comportamento do SoS como um todo pode superar a simples soma dos comportamentos de seus constituintes. Isso revela um paradoxo inerente: a necessidade de projetar e desenvolver um SoS para exibir um comportamento esperado que é, por sua natureza, emergente (Cavalcante; Batista; Oquendo, 2024).

Por fim, a governança e a coordenação em SoS são inerentemente complexas devido à independência dos sistemas constituintes e à frequente ausência de uma autoridade central forte (Darabi; Gorod; Mansouri, 2012). Mecanismos de governança tradicionais podem não ser eficazes nesse cenário. Diferentes formas de controle podem ser aplicadas na aquisição e operação dos sistemas constituintes, e a estrutura de governança deve ser cuidadosamente selecionada e aplicada para garantir um desempenho eficaz. Problemas de desempenho em SoS frequentemente estão ligados a questões de governança não técnicas, como incompatibilidades entre estruturas organizacionais e os requisitos de negócios globais do SoS. Em SoS colaborativos e virtuais, onde não há uma autoridade central com poder coercitivo, a colaboração deve ser voluntária, exigindo que os mecanismos para assegurar essa cooperação sejam projetados no sistema (Tekinerdogan, 2022; Maier, 1998).

As características marcantes de um SoS, como autonomia, comportamentos emergentes e complexidade, requerem abordagens e ferramentas especializadas para uma gestão eficaz. Os SoS apresentam desafios únicos para a governança de dados, tais como a heterogeneidade dos dados, a complexidade das interações e a soluções para manutenção da integridade e veracidade das informações.

2.2 Governança de dados e Proveniência

A governança de dados emerge como um pilar estratégico para as organizações, estabelecendo um conjunto robusto de práticas, processos, funções e estruturas cruciais. Seu objetivo é assegurar que os dados organizacionais sejam gerenciados como ativos de valor estratégico. Essencialmente, a governança de dados delineia as responsabilidades e

os direitos de decisão sobre os dados, especificando quem pode tomar quais ações, com quais informações, quando, sob quais circunstâncias e utilizando quais métodos. Adicionalmente, formaliza políticas, padrões e procedimentos relativos aos dados e monitora sua conformidade, garantindo assim a qualidade, segurança, disponibilidade, integridade e conformidade dos dados em todo o seu ciclo de vida (Caballero; Piattini, 2023).

A implementação eficaz da governança de dados não apenas garante a precisão, consistência e o uso responsável dos dados, mas também auxilia as organizações a atenderem às crescentes exigências legais e regulatórias. Ao melhorar os processos decisórios e proteger informações sensíveis, a governança de dados fortalece a organização (Al-Ruithe; Benkhelifa; Hameed, 2019; Abraham; Schneider; Vom Brocke, 2019). Dados que são bem governados fornecem uma base sólida, confiável e consistente, indispensável para a tomada de decisões em níveis estratégico, tático e operacional. Consequentemente, dados mais confiáveis, oportunos e compreensíveis minimizam riscos e elevam a eficácia organizacional.

Nesse contexto, a proveniência de dados se apresenta como um componente indissociável e essencial da governança de dados. A proveniência refere-se à descrição de um histórico da origem, trajetória e transformações sofridas pelos dados ao longo do tempo. Este rastreamento é essencial para verificar a autenticidade dos dados, determinar responsabilidades e assegurar a integridade contínua das informações utilizadas. Em cenários regulatórios complexos, como os definidos pela Lei Geral de Proteção de Dados (LGPD), a capacidade de demonstrar de forma transparente como os dados foram obtidos, processados e compartilhados é um pilar para garantir a conformidade e mitigar o risco de sanções legais (Singh; Cobbe; Norval, 2018).

Os benefícios da proveniência de dados para a governança de dados estendem-se significativamente ao aumento da confiança nos dados. Quando os dados são utilizados para análises críticas, embasar decisões de negócio ou gerar relatórios, a confiança em sua veracidade e linhagem é primordial. A rastreabilidade proporcionada pela proveniência permite uma identificação clara das fontes e dos processos que contribuíram para um determinado conjunto de dados ou resultado analítico. Essa capacidade não só facilita a validação rigorosa dos dados, como também assegura a reprodutibilidade das análises, um aspecto fundamental para a robustez científica e a tomada de decisão informada (Zhao *et al.*, 2009; Buneman; Khanna; Wang-Chiew, 2001). Em ambientes de nuvem, por exemplo, a proveniência rastreia o uso e a origem dos dados, provando sua autenticidade e permitindo o armazenamento de metadados sobre alterações, o que é fundamental para a

auditabilidade e responsabilização (Ramane; Vasudevan; Allaphan, 2014). Em domínios críticos como o da saúde, a capacidade de rastrear a origem de dados e informações derivadas de múltiplas transformações é um requisito fundamental, especialmente em sistemas que integram dados de diversas fontes (Hardin; Kotz, 2021). Similarmente, em observatórios astronômicos, a proveniência detalhada é chave para demonstrar a qualidade e confiabilidade dos dados, sendo essencial capturá-la de forma minuciosa, registrando o que acontece durante o processamento dos dados (Servillat *et al.*, 2020). A utilização de modelos padronizados, como o W3C PROV, que se baseia nos conceitos de Entidade, Atividade e Agente, é comum para estruturar essa informação de proveniência em diversos contextos (Servillat *et al.*, 2020; Costa *et al.*, 2021).

Portanto, a integração da proveniência de dados aos mecanismos de governança de dados é fundamental para as organizações que almejem excelência na gestão de seus ativos informacionais, assegurando pilares como qualidade, segurança, transparência e responsabilização no tratamento dos dados. Tais elementos são a base para o uso ético, eficiente e estratégico dos dados. A ausência desse componente pode acarretar consequências sérias, comprometendo a capacidade da organização de responder a auditorias, a precisão de suas análises preditivas e descritivas, e, fundamentalmente, a confiança depositada por usuários internos e externos nos seus sistemas de informação. Portanto, cultivar uma cultura de governança enriquecida pela proveniência de dados é um passo decisivo para transformar dados em verdadeiro conhecimento e vantagem competitiva (Caballero; Piattini, 2023; Golan *et al.*, 2022).

2.2.1 PROV

O padrão PROV (Groth; Moreaun, 2013) constitui a recomendação do *World Wide Web Consortium* (W3C) para a representação de dados de proveniência em múltiplos contextos. Ele é um conjunto de normas, modelos e diretrizes que visam padronizar a forma como dados de proveniência são descritos, armazenados e compartilhados entre sistemas heterogêneos (Gil; Miles, 2013). No centro desse conjunto está o PROV-DM (Moreau; Missier, 2013), ou Modelo de Dados PROV. Esse modelo estabelece os principais elementos que compõem a representação da proveniência: (i) entidades, (ii) atividades e (iii) agentes, além dos relacionamentos entre eles.

No contexto do PROV-DM, entidades correspondem a objetos ou dados que possuem estado persistente em um determinado instante; atividades representam os processos, execuções ou ações que geram, utilizam ou modificam entidades; e agentes são os responsáveis

por iniciar, controlar ou supervisionar tais atividades.

O PROV-DM é fundamentalmente construído sobre um conjunto de relações que conectam os elementos de proveniência (Entidades, Atividades, Agentes) e define várias dessas relações para expressar o fluxo de proveniência. Dentre as relações definidas no modelo, destacam-se aquelas que explicitam as dinâmicas de geração, derivação, uso, invalidação e as diversas formas de atribuição e associação de responsabilidade.

A derivação indica que a existência, conteúdo ou características de uma entidade se devem, em parte, a outra entidade, sendo *WasDerivedFrom* a relação principal. O PROV-DM também define o *bundle*, um conjunto nomeado de descrições de proveniência. Considerado como uma entidade, um *bundle* constitui um mecanismo essencial para expressar a proveniência da própria proveniência. Ainda nesse contexto, *MentionOf* (Moreau; Lebo, 2013) é um relacionamento ternário que expressa uma ligação ou derivação entre dois *bundles*, destacando uma conexão mais complexa que envolve múltiplas entidades e atividades.

Para ilustrar o propósito de um *bundle* e da relação *MentionOf*, considere que algumas aplicações podem desejar expandir as descrições de uma entidade $e1$, localizada em um *bundle* bA, com informações adicionais. O desafio é incorporar novas informações sem alterar completamente o original ou criar duplicações desnecessárias. Para resolver esse problema, é possível criar uma nova entidade $e2$, definida como especialização da entidade $e1$ do *bundle* bA, por meio da relação *MentionOf*. Dessa forma, as aplicações que processam $e2$ podem compreender que os atributos de $e2$ foram estabelecidos com base nas descrições de $e1$ no *bundle* bA.

Coleções representam estruturas lógicas que agrupam entidades e, portanto, são também tratadas como entidades passíveis de ter sua proveniência registrada. O relacionamento *hadMember* indica que uma entidade do tipo coleção contém outra entidade como membro.

A relação *WasGeneratedBy* estabelece que uma entidade foi gerada por uma atividade. Complementarmente, Used indica que uma atividade utilizou ou consumiu uma entidade em seu processo. A relação *WasInvalidatedBy* refere-se ao fato de que uma entidade foi invalidada por uma atividade, marcando o fim de sua validade ou existência dentro de um determinado contexto.

A rastreabilidade da influência entre atividades é capturada pela relação *WasInformedBy*, que denota que uma atividade foi influenciada ou informada por outra atividade.

Já a relação *WasDerivedFrom* expressa a derivação de uma entidade a partir de outra, indicando que a existência, conteúdo ou características da entidade derivada se devem, pelo menos em parte, à entidade de origem.

No que tange à responsabilidade, a relação *WasAttributedTo* associa uma entidade a um agente responsável por ela, podendo ser entendida como uma forma concisa de indicar que o agente foi responsável pela atividade que gerou a entidade. A relação *WasAssociatedWith* liga uma atividade a um agente que teve alguma forma de responsabilidade por sua execução. Adicionalmente, *ActedOnBehalfOf* descreve a delegação de responsabilidade, indicando que um agente atuou em nome de outro agente em relação a uma atividade.

2.2.2 PROV-JSON: Serialização em JSON

PROV-JSON (Huynh *et al.*, 2013) é um formato de serialização para dados de proveniência baseado no JSON (um formato leve de intercâmbio de dados). Projetado para ser legível por humanos e facilmente interpretável por máquinas, o PROV-JSON facilita a integração dos dados de proveniência em aplicações web modernas e em sistemas baseados em JSON. O PROV-JSON foi escolhido para ser usado pela abordagem proposta nesse trabalho como formato de representação de dados de proveniência devido ao seu bom desempenho, interoperabilidade, legibilidade e compatibilidade.

2.3 Bases de dados orientadas a grafos

Embora o PROV-JSON represente um avanço na serialização, ele apresenta limitações quando se trata de consultas, especialmente na ausência de suporte direto por parte de sistemas de banco de dados. Em razão disso, torna-se necessário armazenar os dados de proveniência em um banco de dados cujo modelo de dados seja aderente à estrutura conceitual do padrão PROV, de modo a viabilizar consultas eficientes. No contexto da abordagem proposta, os dados de proveniência foram carregados no Neo4j (Neo4j, 2025), um sistema de banco de dados orientado a grafos.

Um banco de dados de grafos modela e persiste informações por meio de uma estrutura composta por nós, relacionamentos (arestas) e propriedades associadas (Anuyah; Bolade; Agbaakin, 2024). Tais sistemas são especialmente utilizados em contextos onde as relações entre elementos de dados são, no mínimo, tão relevantes para a análise quanto suas propriedades (Almeida *et al.*, 2019).

A estrutura do modelo PROV, essencialmente baseada em entidades, atividades, agentes e relações representadas como um grafo, encontra uma correspondência natural em bancos de dados de grafos. Essa aderência dispensa a necessidade de transformações complexas ou mapeamentos adicionais entre o grafo de proveniência, originalmente representado em PROV-JSON, e o modelo interno do banco de dados. O Neo4j permite a persistência direta do grafo de proveniência em sua forma nativa, mantendo a semântica dos relacionamentos e a navegabilidade entre os elementos (Wercelens *et al.*, 2019; Almeida *et al.*, 2019).

2.4 Considerações finais

Este capítulo dedicou-se a construir o alicerce teórico para a compreensão da governança de dados em SoS. Iniciou-se pela caracterização dos SoS, ressaltando sua natureza complexa, distribuída e emergente, fruto da colaboração entre sistemas autônomos e heterogêneos. Foram abordados, ainda, os desafios inerentes a esses ambientes, especialmente no que se refere à governança e coordenação, dada a autonomia gerencial dos sistemas constituintes e a frequente ausência de uma autoridade central. Na sequência, foi aprofundado o papel crucial da governança de dados, entendida como um conjunto de práticas para assegurar que os dados sejam geridos como ativos estratégicos, garantindo sua qualidade, segurança, integridade e conformidade. Dentro desse escopo, a proveniência de dados emergiu como um componente essencial. Ela oferece um histórico detalhado da origem, trajetória e transformações dos dados, sendo fundamental para a rastreabilidade, a verificação de autenticidade e a atribuição de responsabilidades. Enfatizou-se, também, como a proveniência é importante para aumentar a confiança nos dados e para o cumprimento de requisitos regulatórios.

O padrão W3C PROV foi, então, detalhado como uma recomendação para a representação padronizada de informações de proveniência. A análise concentrou-se em seus elementos centrais e nos relacionamentos que descrevem o fluxo de derivação e responsabilidade. O conceito de *bundle* foi introduzido como um mecanismo para encapsular descrições de proveniência e o relacionamento *MentionOf* para conectar informações entre diferentes contextos. A serialização desses dados por meio do PROV-JSON foi mencionada como um formato leve e que favorece a interoperabilidade.

Por fim, analisou-se a pertinência de bancos de dados orientados a grafos para a persistência e consulta de dados de proveniência. A afinidade natural entre a estrutura

em grafo do modelo PROV e o modelo de dados desses bancos, como o Neo4j, simplifica a representação e a execução de consultas complexas sobre o histórico dos dados.

Com esses fundamentos teóricos estabelecidos, o próximo capítulo discute trabalhos relacionados, evidenciando as lacunas existentes que justificam a proposta de uma nova abordagem para governança de dados em SoS.

3 Trabalhos Relacionados

3.1 Monitoramento de SoS usando Proveniência

A literatura reflete a importância do estudo da proveniência em diferentes áreas do conhecimento (Oliveira; Oliveira; Braganholo, 2018). No contexto de governança de dados em SoS, soluções foram propostas para lidar com o monitoramento dos SoS, mesmo que usando dados de proveniência de forma implícita.

A seleção dos trabalhos relacionados apresentados neste capítulo seguiu um processo sistemático, com buscas realizadas em diversas bases de dados científicas, com foco principal no Google Scholar e Scopus. A estratégia de busca foi dividida em duas etapas. Inicialmente, a pesquisa abordou os conceitos fundamentais de forma ampla, utilizando termos gerais como "sistemas-de-sistemas"(*systems of systems*), "proveniência de dados"(*data provenance*) e "governança de dados"(*data governance*) para estabelecer uma base teórica sólida. Posteriormente, a busca foi refinada com combinações de palavras-chave mais específicas para identificar trabalhos diretamente alinhados ao escopo do trabalho, tais como "proveniência em governança de dados"(*provenance + data governance*), "proveniência em sistemas de sistemas"(*provenance + systems of systems*), "monitoramento de sistemas-de-sistemas"(*monitoring of systems of systems*) e variações. O principal critério para a seleção dos artigos foi o número de citações, priorizando trabalhos com maior impacto e reconhecimento na comunidade científica. No entanto, este critério foi flexibilizado para incluir artigos que, apesar de recentes ou com um número menor de citações, apresentavam um contexto julgado particularmente pertinente ou uma abordagem diferenciada, garantindo assim uma visão abrangente e atualizada do estado da arte.

Vierhauser et al. (2016) apresentam o REMINDS (REquirements Monitoring INfrastructure for Diagnosing Systems of Systems), um *framework* flexível para o desenvolvimento de soluções de monitoramento que abrange diferentes sistemas que formam um SoS. O REMINDS é estruturado em camadas para coletar eventos e dados; acumular e persistir esses dados; avaliação de dados como checagem de restrições; e para interfaces

de usuário e visualizações. O *framework* utiliza um modelo de eventos para gerenciar e analisar eventos arbitrários e suporta a instrumentação de sistemas heterogêneos através de *probes*, que podem ser desenvolvidos com base em templates fornecidos. A capacidade de definir e checar restrições sobre eventos e dados, inclusive através de uma *Domain-Specific Language* (DSL), permite a verificação do comportamento do SoS em relação aos seus requisitos. A avaliação do REMINDS demonstrou sua flexibilidade e escalabilidade para monitorar SoS industriais com cargas de eventos realistas, abordando desafios como monitoramento em diferentes camadas, entre sistemas diversos, tecnologias variadas e com diferentes velocidades operacionais.

Complementarmente, Kritzinger et al. (2019) destacam a necessidade de apoio à visualização no monitoramento de SoS, particularmente por meio de estudos com usuários que revelam a eficácia das ferramentas visuais na compreensão do comportamento dos sistemas. A abordagem utiliza o *framework* REMINDS (Vierhauser *et al.*, 2016) e oferece várias possibilidades de visualização que permitem aos usuários monitorar efetivamente o status dos SoS e detectar violações. As capacidades de visualização do REMINDS incluem uma visão geral do sistema baseada em grafos e gráficos para o status das restrições, diagramas de tendência e de intervalo para analisar violações ao longo do tempo, e visualizações de eventos para detalhar a ocorrência e o fluxo de eventos. Um estudo de caso com engenheiros industriais e pesquisadores monitorando um sistema de automação do mundo real confirmou que foi possível monitorar o sistema e diagnosticar violações, considerando as capacidades de visualização essenciais para entender o comportamento de sistemas complexos. Essa abordagem ressalta o papel crítico do design centrado no usuário em sistemas de monitoramento, garantindo que as partes interessadas possam interpretar intuitivamente dados complexos e tomar decisões.

Chreim et al. (2024) exploram o tema por meio de uma abordagem de hipergrafo multinível, abordando os desafios relacionados à reconfiguração e otimização em SoS. Para resolver essa questão, o estudo propõe um novo *framework* denominado *Multi-Level Stochastic Hypergraph* (MLSHG). O *framework* utiliza a estrutura de hipergrafos para representar as relações complexas e hierárquicas entre múltiplos componentes do sistema, diferenciando explicitamente entre componentes com comportamento determinístico e estocástico. O MLSHG captura atributos essenciais como desempenho, funções, capacidades e a própria existência dos componentes, permitindo uma representação fiel da dinâmica do SoS. Para complementar o *framework*, os autores desenvolveram um algoritmo de supervisão que integra monitoramento de detecção de falhas ou a adição de novos componentes, bem como reconfiguração para ajuste e realocação de missões com base nas capacidades

disponíveis para manter o desempenho e alcançar os objetivos de longo prazo. A validação do *framework* foi realizada por meio de um estudo de caso em um SoS de colheita de cogumelos, envolvendo componentes biológicos, humanos e robóticos. Os resultados demonstraram que a abordagem de reconfiguração baseada em capacidade possui um tempo computacional baixo que varia linearmente com o aumento do número de componentes, garantindo a escalabilidade do sistema. O uso de um limiar adaptativo, que é ajustado dinamicamente com base no nível de distúrbios estocásticos do sistema para acionar a reconfiguração permitiu ajustes mais rápidos. A pesquisa enfatiza a necessidade de monitoramento contínuo para avaliar as contribuições dos sistemas constituintes ao desempenho geral do SoS, especialmente em ambientes caracterizados por incerteza e variabilidade.

Singh, Cobbe e Norval (2018) propõem o conceito de proveniência de decisão, que envolve o uso de métodos de proveniência para expor os *pipelines* de decisão: cadeias de entradas, a natureza e os efeitos das decisões e ações tomadas nos sistemas. O objetivo é auxiliar nas considerações de responsabilidade em sistemas algorítmicos de tomada de decisão, particularmente aqueles complexos e interconectados, facilitando a supervisão, auditoria, conformidade, mitigação de riscos e o empoderamento do usuário.

Em sua abordagem, Singh, Cobbe e Norval (2018) argumentam que a proveniência de decisão pode ajudar a entender o comportamento do sistema, gerenciar conformidade e obrigações, apoiar auditorias regulatórias e investigações técnicas, além de investigações legais e de responsabilidade. Um estudo de caso focado em *pipelines* de *Machine Learning* (ML) ilustra como a proveniência pode revelar a natureza dos dados de treinamento identificando vieses ou erros, auxiliar na operação de modelos e facilitar a investigação de decisões tomadas por modelos de ML. Para viabilizar a proveniência de decisão, considerações de implementação incluem mecanismos de captura, gerenciamento dos dados de proveniência, garantia de confiança nos dados de proveniência e formas de tornar esses dados significativos e utilizáveis por diversas partes interessadas através de padrões como W3C PROV e interfaces amigáveis.

Por sua vez, Kong et al. (2020) apresentam uma abordagem que permite observar o comportamento do SoS sem alterar o código-fonte, por meio do monitoramento de rastros de execução. Reconhecendo que as técnicas tradicionais de verificação eram insuficientes para garantir a confiabilidade de software complexo após sua implantação, os autores propuseram uma abordagem de instrumentação invasiva para observar o comportamento

interno do software. O mecanismo utiliza a infraestrutura srcML[1] para analisar o código-fonte e permitir a inserção seletiva de código de monitoramento. Isso possibilita a coleta de dados detalhados do rastro de execução, tais como sequência de chamadas de função, duração e frequência, que são fundamentais para a verificação em tempo de execução e a predição de falhas. O método é particularmente benéfico para manter a integridade dos sistemas monitorados, fornecendo conhecimento sobre seu estado operacional. A capacidade de monitoramento em tempo real é fundamental para identificar problemas e garantir a conformidade com padrões operacionais pré-definidos.

A viabilidade e a performance do *framework* foram validadas através de testes e um estudo de caso com o *software* Nginx[2] e os resultados confirmam que a ferramenta é capaz de instrumentar código complexo e coletar dados de execução com uma sobrecarga de desempenho gerenciável, demonstrando ser uma solução prática para melhorar a confiabilidade do software em ambientes operacionais. Embora o trabalho se concentre nos rastros de um sistema de software único, os princípios de coleta detalhada de dados de execução são relevantes para entender o comportamento dos sistemas componentes dentro de um SoS, fornecendo dados de proveniência sobre sua execução.

Calabro et al. (2021) introduzem o MENTORS (*Monitoring ENvironment FOR Sos*), um ambiente de monitoramento concebido para SoS, com foco na avaliação de novos dispositivos integrados a um sistema existente. O trabalho parte da premissa de que a complexidade e a heterogeneidade dos SoS, frequentemente compostos por componentes de terceiros, geram vulnerabilidades significativas e dificultam a criação de sistemas de monitoramento eficazes devido à falta de um entendimento comum entre os diferentes especialistas envolvidos. Para superar esse desafio, o MENTORS propõe uma ontologia central chamada MONTOLOGY, que formaliza e unifica o conhecimento sobre o SoS e o domínio do monitoramento. O objetivo final do MENTORS é reduzir custos, aumentar a flexibilidade e melhorar o controle de qualidade no monitoramento de SoS, fornecendo um framework que padroniza a especificação de regras e promove a interoperabilidade. A validação da proposta com casos de uso reais é delineada como trabalho futuro.

Para concluir, Krismayer, Rabiser e Grunbacher (2017) abordam o desafio de definir regras de monitoramento, ou restrições, para Sistemas-de-Sistemas (SoS) complexos, com o objetivo de detectar desvios de comportamento. O trabalho propõe minerar automaticamente, a partir de logs de eventos, as restrições necessárias ao monitoramento em tempo de execução, evitando a dependência de conhecimento de domínio profundo

[1] https://www.srcml.org/
[2] https://nginx.org/

geralmente ausente em equipes de desenvolvimento independentes. O método combina técnicas de mineração de especificações, mineração de processos e aprendizado de máquina para extrair diferentes tipos de restrições, tanto temporais (sobre a ocorrência e ordem dos eventos) quanto de valor (sobre os dados associados aos eventos).

A abordagem proposta opera em quatro etapas: inicialmente, identifica sequências frequentes de eventos para sugerir restrições temporais. Em seguida, mapeia dados dos logs em vetores de características visando encontrar restrições simples. Posteriormente, utiliza aprendizado de máquina para extrair regras mais complexas e, por último, classifica as restrições mineradas segundo relevância. A viabilidade foi demonstrada por meio de um estudo de caso com logs reais de um SoS industrial, onde especialistas validaram como úteis 11 das 15 restrições mais relevantes. Os resultados evidenciam que a mineração automática é uma solução prática para o desafio industrial de definir e manter restrições, reduzindo a dependência da especificação manual.

3.2 Considerações Finais

A análise dos trabalhos relacionados revela um panorama diversificado de abordagens para o monitoramento de Sistemas-de-Sistemas (SoS). Soluções como o framework REMINDS (Vierhauser *et al.*, 2016) e MENTORS (Calabro *et al.*, 2021) oferecem modelos e ambientes para o monitoramento em tempo real e avaliação do comportamento dos SoS com foco na detecção de violações de requisitos ou anomalias. O REMINDS se destaca por sua arquitetura em camadas flexível para coletar, persistir e avaliar eventos contra restrições definidas em uma DSL, sendo validado em ambientes industriais complexos. O trabalho de Kritzinger et al. (2019) complementa esta visão, enfatizando a necessidade de visualizações eficazes, como as fornecidas pelo REMINDS, para permitir que engenheiros compreendam o comportamento do sistema e diagnostiquem violações de forma intuitiva. Por sua vez, o MENTORS propõe um ambiente de monitoramento orientado pelo conhecimento, utilizando uma ontologia (MONTOLOGY) para unificar o conhecimento de especialistas e padronizar a definição de regras de monitoramento, que são enriquecidas continuamente por um componente de aprendizagem.

Outras abordagens focam na automação e supervisão em ambientes dinâmicos. O trabalho de Chreim et al. (2024) introduz o modelo MLSHG, baseado em hipergrafos multinível, para a supervisão ativa de SoS, que não apenas monitora, mas também realiza a reconfiguração e otimização do sistema em resposta a comportamentos imprevisíveis.

Em contraste, a abordagem de Krismayer, Rabiser e Grunbacher (2017) enfrenta o desafio da definição de regras, propondo um método que minera automaticamente restrições de monitoramento a partir de logs de eventos, utilizando técnicas de aprendizado de máquina para extrair conhecimento de domínio que muitas vezes não está documentado.

Em um nível mais granular, o trabalho de Kong et al. (2020) apresenta um método para o monitoramento de rastros de execução de software via instrumentação invasiva. Embora centrado em sistemas únicos, seus princípios são relevantes para a obtenção de dados de proveniência sobre a execução de componentes individuais dentro de um SoS. Por fim, o estudo de Singh, Cobre e Norval (2018) foca na responsabilização em sistemas algorítmicos, conceituando a "proveniência de decisão" para rastrear e expor os fluxos de informação e as cadeias de decisão, especialmente em pipelines de aprendizado de máquina.

Embora essas abordagens representem avanços significativos no monitoramento de SoS, cobrindo aspectos como o estado operacional, detecção de falhas, desempenho, conformidade de requisitos e responsabilização de decisões, elas tendem a concentrar seus esforços na análise de eventos de execução, na otimização de desempenho ou no monitoramento de requisitos operacionais e de estado do sistema. A pesquisa de Singg, Cobre e Norval (2018) é a que mais se aproxima da proposta deste trabalho ao enfatizar a importância dos fluxos de dados para a responsabilização, porém seu foco é direcionado para a captura de proveniência de processos de tomada de decisão algorítmica e automatizada, como em *pipelines* de aprendizado de máquina. Observa-se, portanto, uma lacuna no que tange ao monitoramento específico e detalhado das transformações de dados, à rastreabilidade do ciclo de vida da informação e à governança de dados em ambientes distribuídos e heterogêneos de SoS de forma explícita, padronizada e pesquisável.

A abordagem `PROVGov-SoS`, proposta neste trabalho, busca preencher essa lacuna ao focar especificamente na governança de dados em SoS. Em contraste com os trabalhos citados, que abordam o monitoramento do estado do sistema ou requisitos operacionais, a `PROVGov-SoS` utiliza a captura, persistência e consulta de dados de proveniência, estruturados segundo o padrão W3C PROV, como elemento central. Essa ênfase na rastreabilidade de dados permite o registro do histórico completo das transformações aplicadas aos dados dentro dos múltiplos sistemas que compõem o SoS, oferecendo uma base integrada para a análise da trajetória dos dados e para a definição e aplicação de políticas de governança de dados no contexto de SoS.

4 PROVGov-SoS

Dados de proveniência atuam como uma espinha dorsal para a governança de dados em SoSs, com o objetivo de assegurar que tais dados sejam rastreáveis, confiáveis e auditáveis (Moreau; Batlajery *et al.*, 2017). Para alcançar esse propósito, a abordagem `PROVGov-SoS` deve ser capaz de capturar e gerenciar os dados de proveniência em um ambiente distribuído e heterogêneo, característico dos SoSs.

4.1 Abordagem

A abordagem `PROVGov-SoS` se fundamenta em uma arquitetura genérica para a captura de proveniência em um modelo de dados padronizado para representar o ciclo de vida da informação. O uso de dados de proveniência é então usado como eixo para a governança de dados em SoS: captura de eventos de transformação, organização de metadados conforme o padrão W3C PROV e manutenção de informações pesquisáveis para auditoria e explicabilidade. Com isso, promove rastreabilidade em grão fino do fluxo entre sistemas, integrando monitoramento de logs e comportamentos dos componentes para extrair eventos de interesse.

4.1.1 Arquitetura da Solução

A arquitetura da solução proposta é composta por seis componentes que interagem para coletar, estruturar e disponibilizar os dados de proveniência, conforme ilustrado na Figura 2.

A execução da `PROVGov-SoS` tem início em cada um dos sistemas que compõem o SoS (passo 1 na Figura 2), que possuem os métodos responsáveis pelas transformações dos dados. Essas informações são encapsuladas em uma mensagem e encaminhadas ao *Broker de Dados* (passo 2). O *Broker* atua como um componente intermediador de comunicação, sendo responsável por receber as mensagens de requisição relacionadas aos dados de

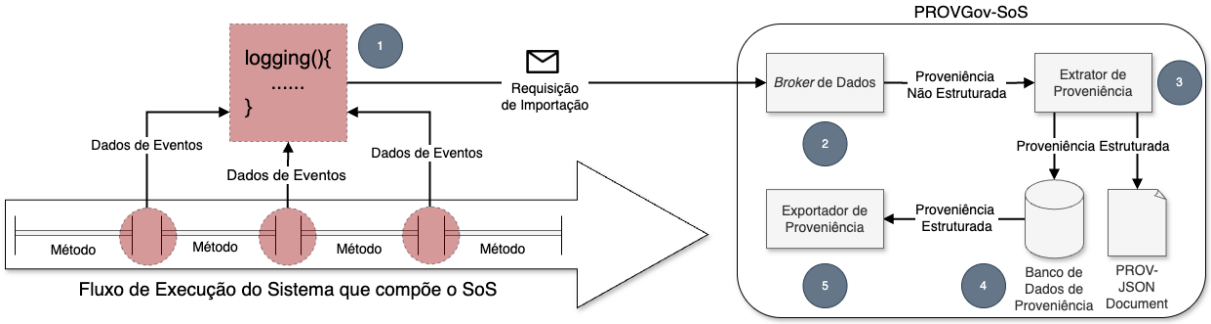


Figura 2: A Arquitetura da Abordagem `PROVGov-SoS`.

proveniência e distribuí-las aos demais componentes da arquitetura da `PROVGov-SoS`. Ele desempenha um papel de orquestrador da coleta de dados de proveniência de forma desacoplada, garantindo a interoperabilidade entre os sistemas participantes e a abordagem de governança proposta.

Uma vez recebida a mensagem de importação de proveniência, o *Broker* encaminha as informações ao *Extrator de Proveniência* (passo 3 na Figura 2). A função deste componente é processar os eventos de transformação de dados. Ele estrutura as informações coletadas em um grafo de proveniência unificado. A construção desse grafo ocorre de forma incremental, consolidando as informações para criar um grafo de proveniência global do SoS. Essa visão unificada permite representar as dependências e o encadeamento entre os dados ao longo de sua trajetória. Uma vez estruturado, o grafo é armazenado em repositórios. Detalhes sobre a construção do grafo são apresentados na Seção 4.1.2.

Uma vez que todos os dados recebidos pelo *Extrator de Proveniência* estejam estruturados em um grafo, eles são armazenados no *Repositório de Documentos* no formato PROV-JSON e o *Banco de Dados de Proveniência* é atualizado (passo 4 na Figura 3). Finalmente, o *Exportador de Proveniência* executa uma série de consultas ao banco, utilizando a linguagem Cypher, para extrair o grafo ou subgrafo relevante para determinada análise (passo 5 na Figura 3). Ao possibilitar consultas baseadas no padrão PROV, aliadas à adoção de um banco de dados orientado a grafos, a `PROVGov-SoS` facilita a análise e interpretação, por parte dos usuários, sobre como os dados foram produzidos e transformados ao longo do tempo no SoS. Além disso, os dados de proveniência, por estarem em conformidade com o padrão PROV, podem ser usados por ferramentas existentes, voltadas à visualização, auditoria e análise de dados (Moreau; Batlajery *et al.*, 2017).

4.1.2 Grafo de Proveniência e Uso de Bundles

A construção do grafo de proveniência é o núcleo da abordagem, e para isso, a *PROVGov-SoS* adota e estende conceitos do padrão W3C PROV. Nessa abordagem, todo artefato persistido é modelado como uma entidade segundo esse padrão, assegurando um ponto único de referência para sua identificação e rastreabilidade. Essas entidades são organizadas em coleções, estruturas lógicas que agrupam objetos semanticamente relacionados e possibilitam sua manipulação como uma unidade informacional.

Para organizar as informações e preservar o contexto de cada transação, a abordagem utiliza o conceito de *bundle* do PROV. No padrão, um *bundle* é descrito como a "proveniência da proveniência", funcionando como uma entidade especial que encapsula um subgrafo de proveniência. Neste trabalho, um bundle é interpretado como uma transação específica ocorrida no SoS.

Cada *bundle*, portanto, contém as operações de transformação de dados ocorridas durante a respectiva transação, proporcionando a granularidade necessária para análises futuras e facilitando a rastreabilidade de dados. Esse conceito é flexível e pode ser adaptado a diferentes domínios de aplicação, oferecendo os blocos conceituais fundamentais para descrever a trajetória dos dados em um SoS. Dentro de um *bundle*, as ações que geram, modificam ou invalidam entidades são modeladas como atividades, uma vez que representam os eventos de transformação dos dados. Assim, o *bundle* garante a preservação do contexto semântico de cada transação, além de possibilitar a composição histórica das interações entre os sistemas participantes. O *bundle* atua como uma cápsula de transação, essencial para a organização lógica e temporal dos elementos.

Um exemplo didático de *bundle* no padrão PROV é apresentado na Figura 3, na qual três *bundles* representam diferentes transações de transformação de dados em um SoS. O *bundle A* descreve a importação da entidade (elipses em amarelo) `Import/A`, que consiste em uma coleção contendo as entidades `class/A` e `activity/A`, relacionadas por meio do relacionamento `hadMember` (mem), conforme definido pelo padrão PROV. O *bundle B* representa uma nova transação que modifica os dados previamente importados na transação anterior na atividade `Update/B` (retângulo azul) e gera o `Import/B`. Por fim, o *bundle C* corresponde a uma operação de exclusão dos dados anteriormente importados. As conexões entre entidades e atividades pertencentes a diferentes *bundles* são estabelecidas por meio do relacionamento chamado `MentionOf` (Moreau; Lebo, 2013). Esse relacionamento permite descrever uma entidade como uma especialização de outra, previamente definida em um *bundle* distinto, possibilitando a continuidade semântica entre transações. Vale

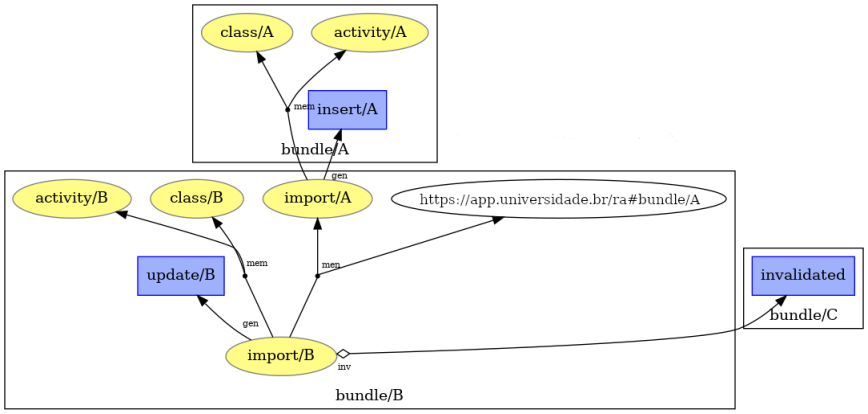


Figura 3: Exemplo de três *bundles* e as ligações de suas entidades usando os relacionamentos *mentionOf* (*mem*) e *wasInvalidatedBy* (*inv*).

destacar que o uso do relacionamento `MentionOf` entre atividades e entidades de *bundles* diferentes é uma contribuição original deste trabalho e não estava previsto na especificação oficial do PROV. Esse novo relacionamento foi incorporado à biblioteca ProvToolbox[1] e será submetido como proposta de contribuição à comunidade.

Dentro dos *bundles*, os relacionamentos desempenham um papel essencial na descrição das interações entre entidades, atividades e agentes. Embora o padrão PROV defina diversos tipos de relacionamentos, a abordagem `PROVGov-SoS` utiliza os seguintes: (i) *wasGeneratedBy*, que associa uma entidade à atividade responsável por sua geração, estabelecendo um vínculo direto entre o dado e seu processo de criação; (ii) *wasInvalidatedBy*, que indica o momento em que uma entidade foi invalidada, destruída ou expirada, como resultado de uma atividade específica; (iii) *hadMember*, utilizado para representar que uma entidade do tipo coleção contém outras entidades como seus membros; e (iv) *MentionOf*, um relacionamento ternário estendido neste trabalho, que expressa a referência ou derivação de uma entidade a partir de outra previamente definida em um *bundle* distinto, possibilitando o encadeamento semântico entre múltiplas transações. O *MentionOf* é particularmente relevante para a construção do grafo de proveniência global do SoS, pois permite conectar informações dispersas entre diferentes componentes do SoS.

4.2 Implementação

O fluxo de trabalho da implementação segue o ciclo de vida da captura de proveniência, desde a interceptação dos eventos nos sistemas de origem até a persistência e consulta dos

[1] https://github.com/lucmoreau/ProvToolbox

dados em um formato estruturado.

O primeiro desafio técnico é capturar as transformações de dados que ocorrem nos sistemas participantes do SoS minimizando a necessidade de alterações em código-fonte. Para isso, é utilizada a Programação Orientada a Aspectos (POA), viabilizada pela biblioteca AspectJ [2]. A POA permite a separação de interesses de captura de proveniência da lógica de negócio principal. Na prática, foram definidos "aspectos"que especificam:

- Pontos de Captura: São as junções no fluxo de execução dos sistemas onde a captura de proveniência deve ocorrer. Foram mapeados métodos-chave responsáveis por operações de criação, atualização e exclusão de dados.
- Ações de Captura: É o código que é executado quando um ponto de captura é atingido. Esse código é responsável por extrair o contexto da operação.

Ao executar um método-alvo, a ação correspondente é acionada, coletando as informações necessárias e as encapsulando em uma mensagem que é enviada ao Broker de Dados. Uma vez que os dados brutos da transformação são recebidos pelo Extrator de Proveniência, eles precisam ser estruturados de acordo com o modelo W3C PROV. Essa etapa é realizada com o auxílio da biblioteca ProvToolbox, uma implementação de referência em Java para manipulação de dados de proveniência. O Extrator de Proveniência utiliza a API da ProvToolbox para:

- Criar Instâncias PROV: Mapear os dados coletados para os conceitos do PROV. Por exemplo, um conjunto de dados importado se torna várias Entidades.
- Construir o Grafo de Transação: Cada transação interceptada é encapsulada em um *bundle*. Dentro desta estrutura, as entidades e atividades são conectadas por relacionamentos PROV, como wasGeneratedBy e wasInvalidatedBy, para formar um subgrafo que descreve atomicamente aquela operação.
- Encadear Transações: Para construir o histórico completo, o relacionamento estendido *MentionOf* é utilizado para conectar entidades entre *bundles* distintos. A biblioteca ProvToolbox foi modificada para suportar essa extensão, permitindo a criação de um grafo de proveniência global e coeso.

[2] https://eclipse.dev/aspectj/

Após a construção do grafo de proveniência com a ProvToolbox, ele é preparado para armazenamento e interoperabilidade. O grafo é serializado para o formato PROV-JSON e mantido em um repositório. Este formato foi escolhido por ser leve, legível por humanos e facilmente processável por máquinas, facilitando a troca de informações entre os componentes da arquitetura e o armazenamento em um repositório de documentos para fins de arquivamento e auditoria. O passo final é a persistência dos dados de proveniência em um formato que permita consultas diretas e eficientes. Para isso, a implementação utiliza o Neo4j (versão 5.26.0), um Sistema Gerenciador de Banco de Dados (SGBD) orientado a grafos. A escolha se deu pela aderência natural entre o modelo conceitual do PROV e o modelo de dados do Neo4j:

- Entidades, Atividades e Agentes do PROV: armazenados como nós no grafo.
- Relacionamentos do PROV (como wasGeneratedBy): armazenados como arestas direcionadas e rotuladas.

Essa correspondência direta elimina a necessidade de mapeamentos objeto-relacionais complexos e preserva a semântica do grafo de proveniência. O componente Exportador de Proveniência utiliza a linguagem de consulta Cypher, nativa do Neo4j, para realizar buscas complexas e travessias no grafo, permitindo análises aprofundadas como as demonstradas no estudo de caso apresentado no Capítulo 5.

4.3 Considerações Finais

Este capítulo apresentou a abordagem PROVGov-SoS, concebida para auxiliar a governança de dados em ambientes de Sistemas de Sistemas por meio da captura, persistência e consulta a dados de proveniência. A arquitetura proposta, baseada em componentes desacoplados como o Broker de Dados e o Extrator de Proveniência, garante a interoperabilidade e a coleta organizada de informações em um ecossistema heterogêneo. O núcleo da abordagem reside na modelagem de dados utilizando o padrão W3C PROV, com destaque para o uso de *bundles* como cápsulas de transação e a extensão do relacionamento *MentionOf* para criar um grafo de proveniência coeso. A implementação, apoiada por tecnologias como AOP e bancos de dados de grafos, oferece uma solução prática e robusta. Com estes fundamentos estabelecidos, o próximo capítulo se dedicará a avaliar a PROVGov-SoS em um estudo de caso real, demonstrando sua aplicabilidade e eficácia.

5 Avaliação

Com o objetivo de avaliar a abordagem `PROVGov-SoS`, foi realizado um estudo de caso utilizando um SoS real, desenvolvido pela Universidade Federal Fluminense (UFF). O sistema, denominado Relatório Anual de Docentes (RAD), é responsável pelo registro anual das atividades acadêmicas dos docentes da universidade. Esse sistema opera em integração com diversos sistemas independentes, tanto internos quanto externos à instituição, consumindo dados que subsidiam a tomada de decisões estratégicas pela Reitoria e pelo MEC.

5.1 Estudo de Viabilidade: O SoS do Relatório Anual de Docentes (RAD)

Os seis sistemas que compõem o SoS RAD são ilustrados na Figura 4 e descritos a seguir: (i) idUFF[1] (*alias* SisAcad), Sistema Acadêmico da Graduação, responsável pela centralização de dados de indivíduos com vínculo vigente ou expirado com a universidade em cursos de graduação; (ii) o CV Lattes[2], sistema de informação mantido pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), que agrega currículos de pesquisadores, estudantes e instituições no Brasil; (iii) o SIAPE[3], sistema do governo federal brasileiro utilizado para gerenciar os registros funcionais dos servidores públicos civis; (iv) o SIGPROJ[4], plataforma destinada ao registro, acompanhamento e avaliação de projetos de extensão em universidades brasileiras; e (v) SISPOS[5] (*alias* SisPG), sistema voltado à gestão de inscrições, estudantes, chamadas públicas, docentes, cursos, pesquisadores, disciplinas e currículos da pós-graduação.

Anualmente, cada docente da universidade deve acessar o sistema RAD para registrar

[1] https://app.uff.br/iduff/
[2] https://lattes.cnpq.br/
[3] https://www.siapenet.gov.br/
[4] http://sigproj.ufrj.br/
[5] http://app.uff.br/sispos/principal

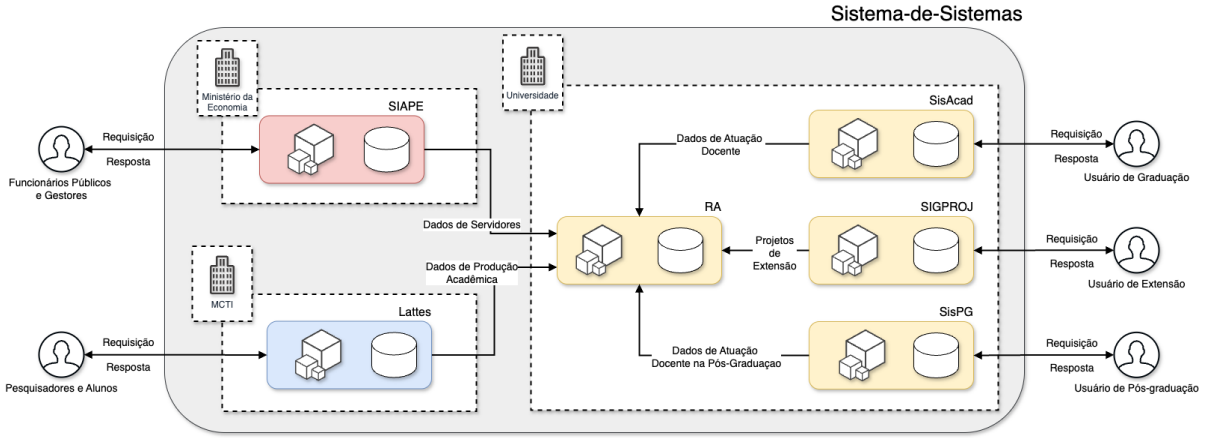


Figura 4: Arquitetura do SoS escolhido para estudo de viabilidade do `PROVGov-SoS`

as atividades acadêmicas realizadas no ano anterior, informando, obrigatoriamente, a carga horária correspondente a cada uma delas. Além das atividades desempenhadas, os produtos gerados, *e.g.*, publicações e orientações, também devem ser incluídos. Com o objetivo de simplificar o processo de preenchimento, o RAD realiza a importação, o ajuste e o processamento de dados provenientes de diversos sistemas. Por exemplo, informações sobre disciplinas ministradas são extraídas do idUFF; produtos acadêmicos, como artigos científicos, são obtidos do CV Lattes; dados sobre projetos são integrados a partir do SIGPROJ; e registros de orientações de alunos da pós-graduação são coletados do SISPOS.

O sistema RAD realiza requisições aos demais sistemas integrantes do SoS por meio de uma fila de mensagens baseada no *RabbitMQ*, a qual gerencia o fluxo de comunicação entre os sistemas participantes. Cada operação de importação/carga iniciada pelo RAD pode desencadear tarefas de inserção, modificação ou mesmo exclusão de dados previamente armazenados. Embora o sistema mantenha registros em forma de *logs* não estruturados sobre inserções e alterações realizadas em suas tabelas, ele não realiza o registro da proveniência dos dados manipulados. Essa limitação compromete a rastreabilidade do caminho de derivação dos dados no âmbito do SoS. Em outras palavras, os *logs* do RAD, além de não consultáveis de forma simples, representam atividades de forma isolada, sem contemplar o ciclo de vida completo dos dados, o que dificulta a análise e a visualização das conexões e transformações que esses dados sofrem ao longo de sua trajetória pelo SoS.

5.2 Alinhamento da Abordagem PROVGov-SoS ao Estudo de Caso

O funcionamento do RAD expõe os desafios de governança que a PROVGov-SoS se propõe a resolver. Atualmente, o RAD mantém registros de logs não estruturados sobre as operações, o que compromete a rastreabilidade do ciclo de vida dos dados. Os logs registram atividades de forma isolada, sem conectar as transformações que um dado sofre ao longo de sua trajetória, tornando a análise de inconsistências uma tarefa complexa e manual.

Nesse contexto, as escolhas de projeto da PROVGov-SoS foram diretamente alinhadas para endereçar essas lacunas:

- Heterogeneidade e Desacoplamento: a natureza distribuída e a variedade de tecnologias dos sistemas do SoS RAD justificam a arquitetura baseada em um Broker de Dados. Essa escolha permite que a captura de proveniência seja implementada de forma desacoplada, sem exigir grandes modificações nos sistemas constituintes.
- Complexidade do Fluxo de Dados: um mesmo dado, como a carga horária de uma disciplina, pode ter sua origem em um sistema, ser importado pelo RAD e, posteriormente, alterado ou invalidado no contexto do SoS. Para rastrear essa trajetória, a modelagem com o padrão W3C PROV e o uso de *bundles* para encapsular transações são essenciais, pois permitem criar um histórico auditável e padronizado.
- Necessidade de Auditoria e Diagnóstico: problemas recorrentes, como registros incorretos, exigem uma capacidade de análise da linhagem dos dados. A escolha de persistir o grafo de proveniência no Neo4j e utilizar a linguagem Cypher oferece uma ferramenta poderosa para executar consultas complexas, permitindo não apenas identificar o estado final de um dado, mas também reconstruir todo o "filme"de suas transformações, conforme será demonstrado nas seções seguintes.

A arquitetura do RAD especificamente inclui componentes centralizadores que orquestram a operação do conjunto:

- Sistemas Constituintes Independentes: O RAD é um SoS porque integra sistemas que são independentes e úteis por si mesmos. SIAPE, Lattes e SISPOS são sistemas autônomos, com seus próprios gestores e objetivos, mas que contribuem com dados para o RAD.

- O Componente Centralizador de Integração: O RAD apresenta um componente que atua como centralizador e integrador. Sua função é consumir os dados heterogêneos vindos de todas as fontes, aplicar as regras de processamento, transformação e validação necessárias para criar o Relatório Anual, que é o comportamento que nenhum dos sistemas componentes poderia alcançar sozinho.

- Propósito e Governança Centralizados: O RAD existe com o propósito de gerar o relatório anual docente, podendo ser categorizado como um sistema-de-sistemas dirigido (Maier, 1998). O objetivo não emerge voluntariamente da interação dos sistemas; ele é dirigido pela necessidade da instituição. A governança do RAD, embora dependa da colaboração dos sistemas fontes, é centralizada no sentido de que o RAD e a lógica de negócio que ele contém ditam como os dados dos outros sistemas serão utilizados para cumprir a missão do SoS.

Em suma, enquanto os sistemas componentes mantêm sua autonomia gerencial e operacional, dentro do contexto do SoS RAD, eles atuam como fontes de dados subordinadas a uma lógica central.

A arquitetura genérica da abordagem, proposta para a governança de dados em SoS, foi instanciada e adaptada para atender às especificidades do RAD. A aplicação da abordagem focou no ponto central de processamento de dados do RAD para capturar a proveniência das transformações de dados.

A abordagem PROVGov-SoS foi projetada para ser minimamente intrusiva, acoplando-se aos mecanismos de comunicação já em uso pelo ecossistema do SoS original. Embora essa abordagem seja funcional, trabalhos futuros podem explorar soluções mais sofisticadas, alinhadas com os desafios de interoperabilidade inerentes aos SoS. No contexto do RAD, que utiliza uma fila de mensagens para gerenciar o fluxo de dados entre os sistemas produtores e consumidor, a PROVGov-SoS se integra nesse ponto de consolidação. Isso significa que a abordagem não impõe uma nova infraestrutura de comunicação entre os sistemas constituintes do SoS, garantindo que sua aplicação seja transparente para a arquitetura pré-existente.

5.3 Modelagem e Implementação

A condução do estudo de viabilidade da PROVGov-SoS no SoS RAD envolveu a modelagem dos dados de proveniência e a implementação da captura desses dados. O pri-

meiro passo consiste na modelagem dos dados de proveniência conforme o PROV. A `PROVGov-SoS` permite a customização de entidades, atividades e agentes do modelo PROV para se adequar ao contexto específico do SoS em análise. No caso do RAD, cada método executado por um sistema participante foi modelado como uma atividade PROV.

Foram definidos três tipos principais de entidades:

- Conjuntos de Dados manipulados: representando coleções que agregam todas as tuplas consumidas ou geradas por uma atividade;
- Atividades/Produtos: entidades que estão associadas a um Conjunto de Dados e contêm atributos mais genéricos, *e.g.*, um afastamento de docente;
- Entidades de extensão: entidades que estendem os tipos de atividades ou produtos com atributos mais específicos, *e.g.*, um determinado tipo de afastamento, carga horária de uma turma ou o evento de publicação de um artigo.

É importante ressaltar que, embora a `PROVGov-SoS` permita a modelagem de cada tupla individual como uma entidade, essa granularidade muito fina geraria um grafo de proveniência excessivamente grande, dificultando sua análise e visualização por parte dos participantes do estudo. Após consulta com especialistas do sistema, foi definido que, para o estudo de viabilidade, a granularidade em nível de conjunto de dados seria adequada e suficiente para atender aos objetivos de rastreabilidade e análise.

A captura dos dados de proveniência nos sistemas que compõem o SoS foi orientada à aplicação (Singh; Cobbe; Norval, 2018) e implementada por meio de programação orientada a aspectos. Todos os pontos de captura foram previamente definidos, e suas respectivas lógicas integradas ao código dos sistemas sob gestão da própria universidade.

No processo do `PROVGov-SoS`, as entidades cruciais são criadas e incorporadas a uma coleção. À medida que novos objetos são gerados durante a execução da lógica, os comportamentos correspondentes são registrados pelo submódulo, com entidades e relacionamentos descritos utilizando o padrão PROV. Para cada operação é então criado um *bundle* específico. O *bundle* recém-criado estabelece uma referência aos *bundles* anteriores sempre que a tarefa está associada a um item já existente. Este mecanismo gera um encadeamento de entidades, permitindo a definição das alterações realizadas ao longo do processo. O uso do relacionamento *mentionOf* desempenhou um papel central nesse processo, contribuindo significativamente para a descrição detalhada do fluxo de dados no contexto do RAD.

5.4 Uso do PROVGov-Sos para auditoria no RAD

Para avaliar a abordagem *PROVGov-SoS* em um contexto prático, este estudo analisou um conjunto de dados que simula o uso real do SoS, abrangendo o período de 2021 a 2024. O conjunto de dados inclui um volume abrangente de informações, como produções acadêmicas, turmas de graduação e pós-graduação, além de registros de docentes e discentes. A avaliação concentrou-se na aplicação da abordagem para investigar problemas de importação frequentemente relatados por usuários finais, visando demonstrar sua utilidade para diagnosticar anomalias do mundo real.

A primeira análise realizada aborda uma inconsistência recorrente observada pelos usuários: o registro da carga horária de uma turma (ch) como zero após determinados processos de importação, o que pode constituir um erro para algumas disciplinas. Para conduzir essa investigação, foram elaboradas consultas na linguagem Cypher [6], executadas diretamente sobre o banco de dados de proveniência, sem a utilização de pacotes de extensão como o APOC[7].

O modelo básico de dados do grafo adotado é apresentado na Figura 5.

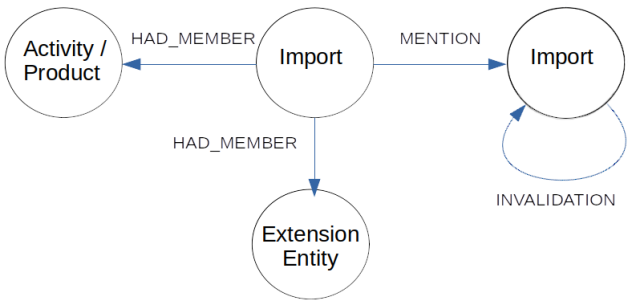


Figura 5: Modelo geral da base de dados em grafos

A estrutura do grafo é detalhada a seguir:

[6] https://neo4j.com/docs/getting-started/cypher/
[7] https://neo4j.com/labs/apoc/

```
{
    "n": {
      "identity": ,
      "labels": [
        "Entity"
      ],
      "properties": {
        "observacao": "",
        "idOrgao": "",
        "name": "",
        "timeInclusao": "",
        "id": "",
        "anoExercicio": "",
        "timeAlteracao": "",
        "matriculaDocente": ""
      },
      "elementId": ""
    }
  }
```


Figura 6: Modelo de uma entidade do tipo "atividade"(*activity*)


```
{
    "n": {
      "identity":,
      "labels": [
        "Entity"
      ],
      "properties": {
        "epochDataImportacao": "",
        "idProduto": "",
        "operacao": "",
        "fonte": "",
        "name": "",
        "idNaFonte": "",
        "importacaoID": "",
        "idAtividade": "",
        "id": "",
        "dataImportacao": "",
        "dataAlteracao": "",
        "dataGeracao": ""
      },
      "elementId": ""
    }
 }
```


Figura 7: Modelo de uma entidade do tipo "importação"(*import*)

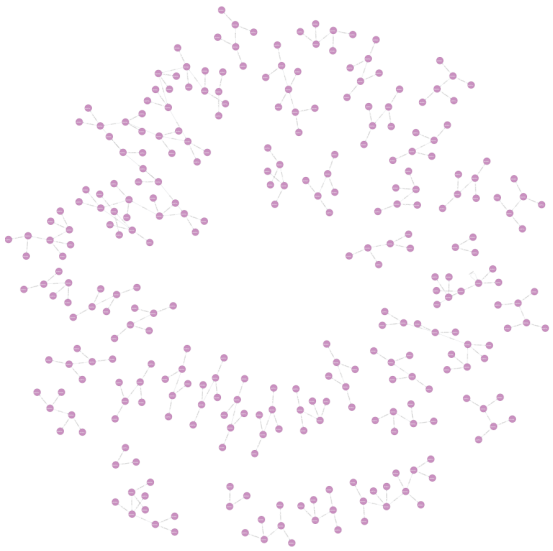
```
{
   "n": {
     "identity": ,
     "labels": [
       "Entity"
     ],
     "properties": {
       "idInstituicao": "",
       "ch": "",
       "tipoAfastamentoId": "",
       "ignoredFields": "",
       "name": "",
       "idAtividade": "",
       "id": ""
     },
     "elementId": ""
   }
}
```


Figura 8: Modelo de uma entidade do tipo "extensão"(*extension*), que complementa outras entidades - neste caso, uma atividade de afastamento

Inicialmente, foi realizada uma consulta (Q1) que seleciona apenas as entidades do grafo de proveniência com carga horária igual a zero e que não sofreram alterações posteriores por nenhum dos sistemas do SoS. Para isso, foram filtrados os nós que não participam de relacionamentos do tipo `Mention`, ou seja, entidades que não foram referenciadas nem modificadas em outras transações (*bundles*). A consulta retorna esses nós e seus relacionamentos (Figura 9), permitindo isolar os dados que representam o estado final do sistema, sem qualquer histórico de atualização ou correção posterior, tanto em formato visual quanto em formato JSON.

```
Q1: MATCH (n {ch:"0"})-[*]-(conectado)
    WHERE NOT EXISTS {(n)<-[:HAD_MEMBER]-()<-[:MENTION]-()}
    RETURN n,conectado
```

Em seguida, foi implementada uma consulta (Q2) para identificar os nomes das disciplinas vinculadas às entidades cujo campo de carga horária se encontra vazio, bem como contabilizar a frequência de ocorrência de cada uma delas. O objetivo dessa consulta é identificar quais disciplinas são mais afetadas por essa inconsistência durante uma importação anual de dados. A resposta da consulta é ilustrada na Tabela 1.




```
[
 {
  "n": {
   "identity": 1,
   "labels": [
    "Entity"
   ],
   "properties": {
    "ch": "0",
    "ignoredFields": "null",
    "localidade": "[Anonimizado]",
    "name": "aulaPosGraduacao/1/1744848646802",
    "disciplinaNome": "BIOESTATÍSTICA I",
    "turmaCodigo": "TM04220221012",
    "qtdAlunos": "7",
    "disciplinaCodigo": "CMM10754",
    "idAtividade": "1",
    "id": "1"
   },
   "elementId": "4:e9180ad6-57e9-4dad-85b7-914df7f5d7b8:1"
  },
  "conectado": {
   "identity": 0,
   "labels": [
    "Entity"
   ],
   "properties": {
```


Figura 9: Resposta da Consulta Q1 - entidades que não sofreram alterações posteriores. (Esquerda) Grafo resultante da consulta; (Direita) Arquivo JSON com resultado da consulta.

```
Q2: MATCH (n {ch:"0"})
    WHERE NOT EXISTS {(n)<-[:HAD_MEMBER]-()<-[:MENTION]-()}
    RETURN n.disciplinaNome AS disciplina, count(*) AS ocorrencias
    ORDER BY ocorrencias DESC
```

disciplina	ocorrencias
DISSERTAÇÃO	187
ESTÁGIO DOCÊNCIA	93
TESE	83
SEMINÁRIO DE ACOMPANHAMENTO DE PROJETOS II	71
DISSERTAÇÃO I	71
SEMINÁRIO DE ACOMPANHAMENTO DE PROJETOS I	62
EXAME DE QUALIFICAÇÃO	52
DISSERTAÇÃO II	43

Tabela 1: Resposta da Consulta Q2 - disciplinas vinculadas a entidades com carga horária vazia.

A terceira consulta (Q3) tem como propósito aprofundar a análise dos dados de proveniência ao investigar os relacionamentos de dependência entre entidades que, embora não estejam diretamente conectadas no grafo, compartilham vínculos indiretos no grafo de proveniência (*i.e.*, fecho transitivo). Essa abordagem permite rastrear possíveis encadeamentos de atualizações que os dados possam ter sofrido após a sua importação inicial para o sistema. Especificamente, a consulta busca identificar diferenças entre versões distintas de uma mesma entidade, com o intuito de detectar inconsistências que possam indicar que certos dados foram inicialmente importados de forma equivocada e, posteriormente, sofreram modificações. A análise dessas discrepâncias ajuda a compreender falhas no

processo de integração de dados entre os sistemas do SoS, e fornece subsídios importantes para a adoção de medidas corretivas e preventivas. A resposta da consulta é mostrada na Tabela 2.

```
Q3: MATCH (n {ch:"0"})<-[*..3]->(m)
    WHERE n.disciplinaNome IS NOT NULL AND m.disciplinaNome IS NOT NULL
    WITH n, m, [propriedade IN keys(n)
      WHERE (n[propriedade] <> m[propriedade] AND propriedade <> 'name')
     ] AS diffs
    WHERE size(diffs)>0
    RETURN n.name, m.name, diffs
    ORDER BY size(diffs) DESC
```

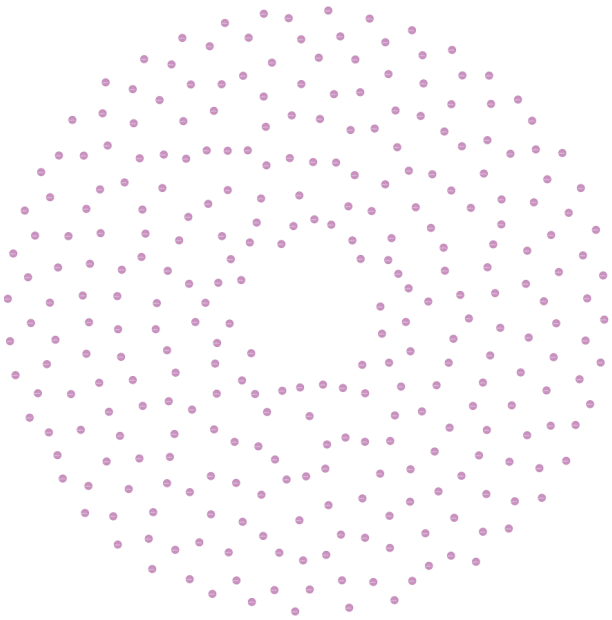
n.name	m.name	diffs
aulaPosGraduacao/4476/1744893030548	aulaPosGraduacao/4476/1744893030541	[ch, qtdAlunos]
aulaPosGraduacao/5015/1744893055406	aulaPosGraduacao/5015/1744893055415	[ch, qtdAlunos]
aulaPosGraduacao/5016/1744893055444	aulaPosGraduacao/5016/1744893055451	[ch, qtdAlunos]
aulaPosGraduacao/5611/1744893108045	aulaPosGraduacao/5611/1744893108039	[qtdAlunos, ch]
aulaPosGraduacao/30/1744848648811	aulaPosGraduacao/30/1744848648828	[qtdAlunos]
aulaPosGraduacao/30/1744848648828	aulaPosGraduacao/30/1744848648811	[qtdAlunos]
aulaPosGraduacao/42/1744848649254	aulaPosGraduacao/42/1744848649276	[qtdAlunos]

Tabela 2: Resposta da Consulta Q3 - versões de entidades que sofreram alterações posteriores a carga com problemas.

Para concluir a análise dos problemas relacionados à carga horária, a próxima consulta (Q4) retorna os nós que destoam do padrão observado para a disciplina na propriedade carga horária, sinalizando um possível erro de registro. O procedimento seleciona os nós em suas versões mais recentes e os agrupa por disciplina, analisando a distribuição dos valores registrados na carga horária para detectar possíveis anomalias. Esse mecanismo de detecção, fundamentado na análise da proveniência e identificação de padrões, oferece possibilidades valiosas para a auditoria do processo de importação. O resultado está representado na Figura 10.

```
Q4: MATCH (n)
    WHERE NOT EXISTS {(n)<-[:HAD_MEMBER]-()<-[:MENTION]-()}
    AND n.disciplinaNome IS NOT NULL AND n.ch IS NOT NULL
    WITH n.disciplinaNome AS disciplina, collect(n.ch) AS cargaHoraria,
       collect(n) AS nosColetados
    WITH disciplina, nosColetados,
       size([ch IN cargaHoraria WHERE ch = "0"]) AS zeros
    WHERE toFloat(zeros) / toFloat(size(cargaHoraria)) < 0.5
    UNWIND nosColetados AS nos
```

```
WITH nos WHERE nos.ch="0"
RETURN nos
```




```
[
 {
  "nos": {
   "identity": 122,
   "labels": [
    "Entity"
   ],
   "properties": {
    "ch": "0",
    "ignoredFields": "null",
    "localidade": "[Anonimizado]",
    "name": "aulaPosGraduacao/19/1744848648099",
    "disciplinaNome": "ESTUDO INDEPENDENTE II",
    "disciplinaCodigo": "INF10014",
    "turmaCodigo": "TM09820221004",
    "qtdAlunos": "1",
    "idAtividade": "19",
    "id": "19"
   },
   "elementId": "4:e9180ad6-57e9-4dad-85b7-914df7f5d7b8:122"
  }
 },
 {
  "nos": {
   "identity": 10931,
   "labels": [
    "Entity"
   ],
   "properties": {
```


Figura 10: Resposta da Consulta Q4 - nós com valores de carga horária fora do padrão. (Esquerda) Grafo resultante da consulta; (Direita) Arquivo JSON com resultado da consulta.

A consulta também poderia ser ajustada para filtrar entidades que sofreram alterações, verificando se houve mudança no campo de carga horária e quando ela ocorreu. Essa análise permite rastrear as etapas do processo de importação que introduziram o erro, possivelmente direcionando a investigação à qualidade dos dados na fonte ou a possíveis falhas na lógica de transformação aplicada durante a ingestão no SoS.

A utilidade do modelo de dados de proveniência transcende a análise de inconsistências pontuais, revelando-se uma ferramenta valiosa para gerar *insights* e realizar consultas sobre a dinâmica de uso dos serviços pelos usuários. Essa abordagem efetivamente transforma a base de dados de uma representação estática de um estado (um "retrato") para um registro dinâmico e histórico (um "filme"), que documenta todo o ciclo de vida da informação. Para ilustrar essa versatilidade, a próxima análise é direcionada para registros de afastamentos de docentes, onde a avaliação dos grafos de proveniência permitiu observar um cenário de alterações de dados completamente distinto.

Para investigar esse novo contexto, foi elaborada uma consulta (Q5) com o objetivo de identificar entidades que sofreram alterações. A execução desta consulta revelou um padrão de alta frequência de modificações para os dados de afastamento evidenciado no grafo pela formação de longas cadeias, como ilustrado na Figura 11. Cada nó em uma

dessas cadeias representa uma nova versão da mesma entidade subjacente, indicando um histórico de sucessivas atualizações ou correções para o mesmo registro. A mesma consulta, quando executada no contexto da Atividade Aula de Pós-Graduação, apresenta um resultado significativamente diferente.

```
Q5: MATCH (n)-[*]-(conectado)
    WHERE EXISTS {(n)-[:MENTION]-()}
    AND n.fonte =~ 'siape/afastamento'
    RETURN n, conectado
    LIMIT 200
```

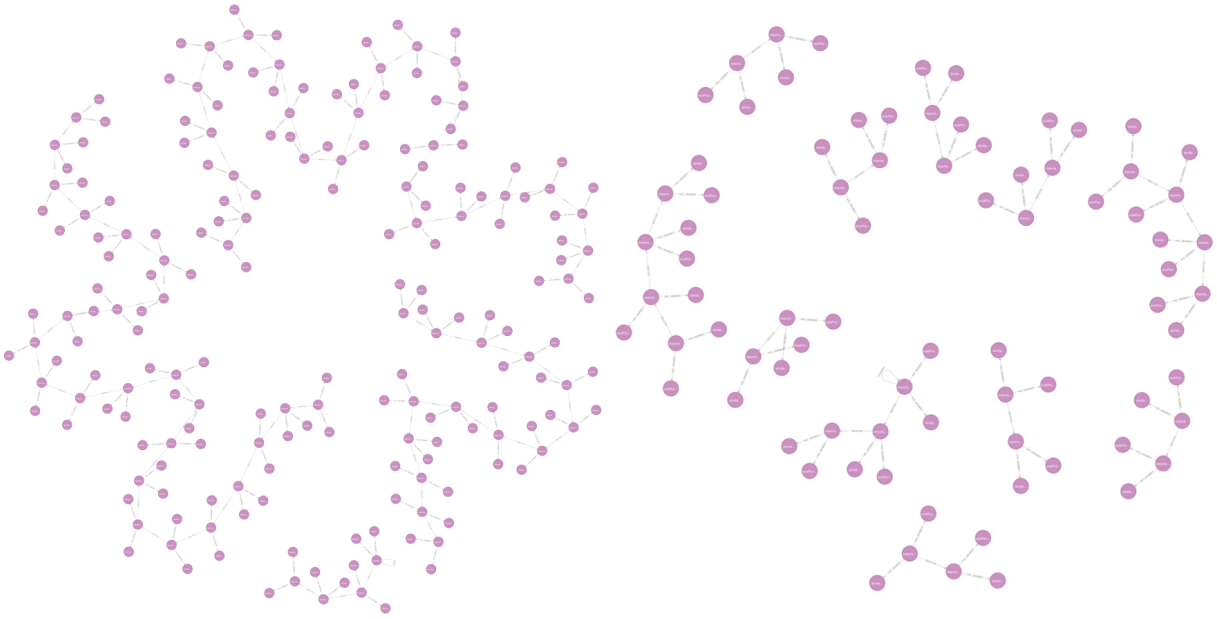


Figura 11: Resposta da Consulta Q5 - busca por nós que sofreram qualquer tipo de alteração, no contexto da Atividade de Afastamento (esquerda) e Aula de Pós-Graducação (direita)

Aprofundando a investigação, a consulta (Q6) foi desenvolvida para classificar os Órgãos com o maior número de exclusões, ou invalidações, de dados importados para o RAD. Essa análise é útil para a equipe técnica, pois, ao invés de tratar erros de forma reativa e individual, permite identificar as fontes mais recorrentes de dados invalidados. A Tabela 3 apresenta os quatro primeiros resultados da consulta.

```
Q6: MATCH (n)-[*]-(conectado)
    WHERE EXISTS {(n)-[:invalidated]->(n)}
    AND conectado.idOrgao <> "null"
    RETURN conectado.idOrgao AS orgao, count(*) AS quantidade
    ORDER BY quantidade DESC
```

orgao	quantidade
473	254
614	98
984	78
10461	76

Tabela 3: Resposta da Consulta Q6 - busca por nós que sofreram invalidação, agrupada por Órgão.

Essas verificações fornecem subsídios práticos para as equipes de suporte e governança de dados. Munida dessa informação, a equipe pode conduzir uma análise aprofundada para determinar se a alta recorrência de alterações para um mesmo indivíduo é (i) um comportamento esperado e pertinente ao processo de negócio, (ii) necessidade de capacitação adequada dos usuários responsáveis pela inserção, validação e manutenção dos dados, ou (iii) um indicativo de falhas sistêmicas, seja na usabilidade de um sistema componente, que induz a correções manuais frequentes, ou em sua lógica operacional, que pode estar gerando dados incorretos. Além disso, pode direcionar seus esforços de forma proativa, focando em corrigir a causa raiz dos problemas junto aos setores mais afetados, otimizando a qualidade dos dados desde sua origem.

A avaliação evidenciou a viabilidade da abordagem `PROVGov-SoS` na identificação de padrões e na realização de análises a partir dos dados de proveniência capturados. A utilização de um banco de dados de proveniência implementado sobre o Neo4J demonstrou ser adequada para esse tipo de aplicação, dado o caráter dos dados envolvidos. Bancos de dados orientados a grafos oferecem mecanismos nativos para consultas complexas, como a busca por caminhos entre nós (como na consulta Q3) e a detecção de padrões recorrentes ou atípicos, características estas que são indispensáveis para aplicações voltadas à identificação de anomalias em sistemas distribuídos, como os SoSs (Anuyah; Bolade; Agbaakin, 2024). Como resultado, a aplicação da proveniência contribuiu para a garantia da qualidade dos dados e para o fortalecimento dos mecanismos de governança no contexto de SoS, ao proporcionar uma visão integrada, auditável e confiável sobre o ciclo de vida dos dados no SoS.

6 Conclusão

Este trabalho propõe a captura de dados de proveniência como elemento central no apoio à governança de dados em SoSs. Para alcançar esse objetivo, foi desenvolvido um conjunto de entidades, atividades e relacionamentos de proveniência, fundamentado no modelo de referência W3C PROV. Esses dados são disponibilizados tanto em formato PROV-JSON quanto em um banco de dados orientado a grafos, que preserva a estrutura nativa dos dados de proveniência. Essa representação padronizada permite que usuários de um SoS possam rastrear o histórico completo de transformações aplicadas a um determinado dado. Tal rastreamento é viabilizado por meio da vinculação explícita entre as transações realizadas pelos diferentes sistemas que compõem o SoS, encapsuladas em estruturas conceituais denominadas *bundles*. Ademais, a adoção de bancos de dados orientados a grafos amplia a capacidade analítica da abordagem, permitindo identificar padrões, inconsistências ou anomalias nos dados processados (Anuyah; Bolade; Agbaakin, 2024). Como resultado, a abordagem `PROVGov-SoS` também se revela promissora para fins de auditoria, promovendo uma cultura de confiança, transparência e rastreabilidade em contextos organizacionais que se apoiam fortemente em dados para a tomada de decisão.

Embora o monitoramento de SoSs com foco em dados de proveniência traga benefícios para a governança e auditoria dos dados, é importante ressaltar que sua implementação não está isenta de desafios. Capturar, estruturar e gerenciar dados de proveniência em um ambiente distribuído e heterogêneo impõe complexidades técnicas, especialmente diante da diversidade de sistemas, formatos de dados e fluxos de processamento. A abordagem `PROVGov-SoS` foi avaliada por meio de um estudo de viabilidade conduzido em um SoS real operado pela UFF. Os resultados demonstraram que a solução é eficaz para apoiar atividades de auditoria, permitindo, por exemplo, a identificação de inconsistências na carga horária registrada em dados importados, um problema recorrente relatado por usuários finais do SoS avaliado.

Como resultados concretos, esse trabalho gerou:

- Uma implementação da *PROVGov-SoS* no contexto do RAD, disponível publicamente no repositório do GitHub em https://github.com/dew-uff/PROVGov-SoS/.
- Proposta de aplicação do relacionamento *MentionOf* para a rastreabilidade transacional de dados. A especificação PROV propõe esta relação com a finalidade ampla de interligar descrições de proveniência que se encontram em *bundles* distintos, sem implicar ordem. Na metodologia da *PROVGov-SoS*, contudo, este relacionamento é utilizado de forma mais estrita e especializada: para estabelecer uma cadeia cronológica de eventos que operam sobre uma mesma entidade de dados, permitindo assim a reconstrução de seu histórico de modificações através de múltiplas transações. Esse relacionamento foi incorporado à biblioteca ProvToolbox (Moreau, 2013) e está disponível publicamente no repositório https://github.com/dew-uff/ProvToolbox-mentionOf.
- Um artigo aceito no Simpósio Brasileiro de Banco de Dados (Monçôres; Braganholo; Oliveira, 2025).

Como trabalhos futuros, destaca-se a implementação de funcionalidades de visualização e navegação interativa sobre os grafos de proveniência gerados, com o objetivo de tornar as análises mais intuitivas para os usuários. Pretende-se também experimentar diferentes níveis de granularidade na captura de proveniência. Adicionalmente, novas investigações podem explorar abordagens de comunicação mais sofisticadas entre os componentes da `PROVGov-SoS`.

Outra vertente de pesquisa consiste em explorar a interoperabilidade com ferramentas de terceiros, validando na prática o benefício do uso de padrões. O repositório de documentos no formato PROV-JSON foi projetado para ser interoperável, e um trabalho futuro seria integrar a saída da PROVGov-SoS com ferramentas de análise e visualização já existentes que consomem o formato, demonstrando a utilidade da abordagem em um ecossistema de ferramentas mais amplo. Adicionalmente, planeja-se investigar técnicas de coleta de proveniência menos intrusivas e mais eficientes, especialmente em cenários onde a instrumentação do código-fonte não é viável. Inspirado em trabalhos relacionados, como o de Kong et al. (2020), poderiam ser explorados métodos para inserção seletiva de código de captura de eventos de transformação de dados com interferência mínima sobre os sistemas componentes.

Por fim, a aplicação da `PROVGov-SoS` em outros domínios além do contexto acadêmico, aliada ao desenvolvimento de mecanismos analíticos para a detecção automatizada de

anomalias e apoio à tomada de decisão, são opções de trabalhos futuros.

Em conclusão, a governança de dados em SoS pode ser reforçada pela adoção da proveniência de dados aliada ao padrão PROV da W3C. Essa metodologia não apenas simplifica a auditoria e assegura a conformidade regulatória, mas também promove uma cultura de confiança e transparência em ambientes de decisão orientados por dados.

REFERÊNCIAS

ABRAHAM, Rene; SCHNEIDER, Johannes; VOM BROCKE, Jan. Data governance: A conceptual framework, structured review, and research agenda. **International journal of information management**, Elsevier, v. 49, p. 424–438, 2019.

ALLEN, M David *et al.* Provenance for collaboration: Detecting suspicious behaviors and assessing trust in information. *In:* IEEE. INTERNATIONAL Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom). [*S. l.: s. n.*], 2011. p. 342–351.

ALMEIDA, Rodrigo *et al.* Managing data provenance for bioinformatics workflows using AProvBio. **International Journal of Computational Biology and Drug Design**, Inderscience Publishers (IEL), v. 12, n. 2, p. 153–170, 2019.

AMMARA, Umme *et al.* Smart cities from the perspective of systems. **Systems**, MDPI, v. 10, n. 3, p. 77, 2022.

ANUYAH, Sydney; BOLADE, Victor; AGBAAKIN, Oluwatosin. Understanding graph databases: a comprehensive tutorial and survey. **arXiv preprint arXiv:2411.09999**, 2024.

BUNEMAN, Peter; KHANNA, Sanjeev; WANG-CHIEW, Tan. Why and where: A characterization of data provenance. *In:* SPRINGER. DATABASE Theory—ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings 8. [*S. l.: s. n.*], 2001. p. 316–330.

CABALLERO, Ismael; PIATTINI, Mario. **Data governance: From the fundamentals to real cases**. [*S. l.*]: Springer Nature, 2023.

CALABRO, Antonello *et al.* MENTORS: Monitoring Environment for System of Systems. *In:* WEBIST. [*S. l.: s. n.*], 2021. p. 291–298.

CAVALCANTE, Everton; BATISTA, Thais; OQUENDO, Flavio. Looking back and forward: A retrospective and future directions on software engineering for systems-of-systems. **Journal of Software: Evolution and Process**, Wiley Online Library, v. 36, n. 10, e2697, 2024.

CHREIM, Abbass *et al.* Towards Supervision of Stochastic System of Systems Engineering: A Multi-Level Hypergraph Approach. **IEEE Access**, IEEE, 2024.

COSTA, Gabriella Castro Barbosa *et al.* Design, Application and Evaluation of PROV-SwProcess: A PROV extension Data Model for Software Development Processes. **Journal of Web Semantics**, Elsevier, v. 71, p. 100676, 2021.

CURRY, Edward; SCERRI, Simon; TUIKKA, Tuomo. **Data spaces: design, deployment and future directions**. [*S. l.*]: Springer Nature, 2022.

CURRY, Edward; SHETH, Amit. Next-generation smart environments: From system of systems to data ecosystems. **IEEE Intelligent Systems**, IEEE, v. 33, n. 3, p. 69–76, 2018.

DARABI, Hamid R; GOROD, Alex; MANSOURI, Mo. Governance mechanism pillars for systems of systems. *In:* IEEE. 2012 7th International Conference on System of Systems Engineering (SoSE). [*S. l.: s. n.*], 2012. p. 374–379.

FU, Xin *et al.* Data governance in predictive toxicology: A review. **Journal of cheminformatics**, Springer, v. 3, p. 1–16, 2011.

GAMMACK, David; SCOTT, Steve; CHAPMAN, Adriane P. Modelling provenance collection points and their impact on provenance graphs. *In:* SPRINGER. INTERNATIONAL Provenance and Annotation Workshop (IPAW). [*S. l.: s. n.*], 2016. p. 146–157.

GEORGE, Jomon; SANTHANAKRISHNAN, T *et al.* System of systems architecture for generic torpedo defence system for surface ships. **Advances in Military Technology**, v. 14, n. 2, p. 307–319, 2019.

GIL, Yolanda; MILES, Simon. **PROV Model Primer**. [*S. l.: s. n.*], 2013. https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/.

GOLAN, Jacob *et al.* Benefit sharing: Why inclusive provenance metadata matter. **Frontiers in Genetics**, Frontiers Media SA, v. 13, p. 1014044, 2022.

GROTH, Paul; MOREAUN, Luc. **PROV Overview**. [*S. l.: s. n.*], 2013. https://www.w3.org/TR/prov-overview/. Accessado em 24 Abr. 2023.

HARDIN, Taylor; KOTZ, David. Amanuensis: Information provenance for health-data systems. **Information Processing & Management**, Elsevier, v. 58, n. 2, p. 102460, 2021.

HERSCHEL, Melanie; DIESTELKÄMPER, Ralf; BEN LAHMAR, Houssem. A survey on provenance: What for? What form? What from? **VLDB J.**, v. 26, n. 6, p. 881–906, 2017. DOI: 10.1007/S00778-017-0486-1. Disponível em: https://doi.org/10.1007/s00778-017-0486-1.

HUYNH, Trung Dong *et al.* **PROV-JSON Serialization**. [*S. l.: s. n.*], 2013. https://www.w3.org/submissions/prov-json/. Accessado em 24 Abr. 2023.

IBNE HOSSAIN, Niamat Ullah *et al.* Modeling and assessing cyber resilience of smart grid using Bayesian network-based approach: a system of systems problem. **Journal of Computational Design and Engineering**, Oxford University Press, v. 7, n. 3, p. 352–366, 2020.

KONG, Shiyi *et al.* Runtime monitoring of software execution trace: Method and tools. **IEEE Access**, IEEE, v. 8, p. 114020–114036, 2020.

KRISMAYER, Thomas; RABISER, Rick; GRUNBACHER, Paul. Mining constraints for event-based monitoring in systems of systems. *In:* IEEE. 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE). [*S. l.: s. n.*], 2017. p. 826–831.

KRITZINGER, Lisa Maria *et al.* A user study on the usefulness of visualization support for requirements monitoring. *In:* IEEE. WORKING Conference on Software Visualization (VISSOFT). [*S. l.: s. n.*], 2019. p. 56–66.

LEE, Oon Ling *et al.* A smart city transportation system of systems governance framework: a case study of Singapore. *In:* IEEE. 2019 14th Annual Conference System of Systems Engineering (SoSE). [*S. l.: s. n.*], 2019. p. 37–42.

LIS, Dominik; OTTO, Boris. Data governance in data ecosystems–insights from organizations. *In:* AMERICAS Conference on Information Systems (AMCIS). [*S. l.: s. n.*], 2020.

MAIER, Mark W. Architecting principles for systems-of-systems. **Systems Engineering: The Journal of the International Council on Systems Engineering**, Wiley Online Library, v. 1, n. 4, p. 267–284, 1998.

MONÇÔRES, Jéssica; BRAGANHOLO, Vanessa; OLIVEIRA, Daniel. Governança de Dados em Sistemas-de-Sistemas: Uma Abordagem Orientada à Dados de Proveniência. *In:* SIMPÓSIO Brasileiro de Banco de Dados (SBBD). [*S. l.: s. n.*], 2025.

MOREAU, Luc. **ProvToolbox. Java library to create and convert W3C PROV data model representations**. [*S. l.: s. n.*], 2013. https://lucmoreau.github.io/ProvToolbox/. Accessado em 24 Abr. 2023.

MOREAU, Luc; BATLAJERY, Belfrit Victor *et al.* A templating system to generate provenance. **IEEE Transactions on Software Engineering**, IEEE, v. 44, n. 2, p. 103–121, 2017.

MOREAU, Luc; LEBO, Timothy. **PROV-LINKS**. [*S. l.: s. n.*], 2013. https://www.w3.org/TR/2013/NOTE-prov-links-20130430/. Accessado em 24 Abr. 2023.

MOREAU, Luc; MISSIER, Paolo. **PROV-DM: The PROV Data Model**. [*S. l.*], 2013. Accessado em 24 Abr. 2023.

NEO4J. **Neo4j Graph Database**. [*S. l.: s. n.*], 2025. https://neo4j.com/product/neo4j-graph-database. Accessado em 12 Mar. 2025.

OLIVEIRA, Wellington Moreira de; OLIVEIRA, Daniel de; BRAGANHOLO, Vanessa. Provenance Analytics for Workflow-Based Computational Experiments: A Survey. **ACM Comput. Surv.**, v. 51, n. 3, 53:1–53:25, 2018. DOI: 10.1145/3184900. Disponível em: https://doi.org/10.1145/3184900.

RAMANE, Muralikrishnan; VASUDEVAN, Balaji; ALLAPHAN, Sathappan. A provenance-policy based access control model for data usage validation in cloud. **arXiv preprint arXiv:1411.1933**, 2014.

AL-RUITHE, Majid; BENKHELIFA, Elhadj; HAMEED, Khawar. A systematic literature review of data governance and cloud data governance. **Personal and Ubiquitous Computing**, Springer, v. 23, p. 839–859, 2019.

SAGE, Andrew P; CUPPAN, Christopher D. On the systems engineering and management of systems of systems and federations of systems. **Information knowledge systems management**, SAGE Publications Sage UK: London, England, v. 2, n. 4, p. 325–345, 2001.

SERVILLAT, Mathieu *et al.* Towards a provenance management system for astronomical observatories. *In:* SPRINGER. INTERNATIONAL Provenance and Annotation Workshop. [*S. l.: s. n.*], 2020. p. 244–249.

SINGH, Jatinder; COBBE, Jennifer; NORVAL, Chris. Decision provenance: Harnessing data flow for accountable systems. **IEEE Access**, IEEE, v. 7, p. 6562–6574, 2018.

TEKINERDOGAN, Bedir. Obstacles of System-of-Systems. *In:* IEEE. 2022 IEEE International Symposium on Systems Engineering (ISSE). [*S. l.: s. n.*], 2022. p. 1–7.

USLAR, Mathias *et al.* Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: A European perspective. **Energies**, MDPI, v. 12, n. 2, p. 258, 2019.

VIERHAUSER, Michael *et al.* ReMinds: A flexible runtime monitoring framework for systems of systems. **Journal of Systems and Software**, Elsevier, v. 112, p. 123–136, 2016.

WERCELENS, Polyane *et al.* Bioinformatics workflows with nosql database in cloud computing. **Evolutionary Bioinformatics**, SAGE Publications Sage UK: London, England, v. 15, p. 1176934319889974, 2019.

WU, Qin; WU, Min; SUN, Yunzhou. Management analysis of the logistics support system-of-systems of US aircraft carrier formation. *In:* EDP SCIENCES. E3S Web of Conferences. [*S. l.: s. n.*], 2021. v. 253, p. 02021.

ZHAO, Jun *et al.* Linked data and provenance in biological data webs. **Briefings in bioinformatics**, Oxford University Press, v. 10, n. 2, p. 139–152, 2009.