Governança de Dados em Sistemas-de-Sistemas: Uma Abordagem Orientada à Dados de Proveniência*

Jessica Monçôres de Almeida[1], **Vanessa Braganholo**[1], **Daniel de Oliveira**[1]

[1]Instituto de Computação – Universidade Federal Fluminense (IC/UFF)

jmoncores@id.uff.br, {vanessa,danielcmo}@ic.uff.br

Resumo. *O desenvolvimento de Sistemas-de-Sistemas (SoS), que integram sistemas independentes por meio de fluxos claros de dados, tem crescido nos últimos anos. Apesar de vantagens como reúso e resiliência, SoSs enfrentam desafios na governança de dados, especialmente na ausência de mecanismos para controlar o ciclo de vida dos dados. Em SoSs, dados gerados por um sistema são usados por outros, dificultando a garantia de rastreabilidade, qualidade e integridade desde a coleta até o armazenamento. Este artigo propõe a* `PROVGov-SoS`*, uma abordagem de governança baseada na gerência de dados de proveniência. A proposta estrutura o fluxo de informações entre sistemas, permitindo que usuários compreendam o ciclo de vida dos dados no SoS. A abordagem foi avaliada em um estudo de viabilidade em um SoS real, com resultados promissores.*

Abstract. *The development of Systems-of-Systems (SoS), which integrate independent systems through clear data flows, has grown in recent years. Despite advantages such as reuse and resilience, SoSs face challenges in data governance, especially due to the lack of mechanisms to control the data lifecycle. In SoSs, data generated by one system is often used by other systems, making it difficult to ensure traceability, quality, and integrity from collection to storage. This article proposes* `PROVGov-SoS`*, a governance approach based on the management of provenance data. The proposal structures the information flow between systems, allowing users to understand the data lifecycle within the SoS. The approach was evaluated through a feasibility study in a real SoS, showing promising results.*

1. Introdução

Na última década, tem-se observado um aumento no desenvolvimento dos chamados Sistemas-de-Sistemas (SoS) [Maier 1998, Cavalcante et al. 2024]. Os SoSs são formados pela integração de múltiplos sistemas de informação autônomos, cuja interoperabilidade é viabilizada por meio da especificação de fluxos de dados bem definidos (*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 et al. 2024]. Nesse contexto, um SoS distingue-se ao possuir uma natureza colaborativa onde cada sistema que o compõe mantém a capacidade de operar de forma independente, mas também são capazes de atuar junto com os demais, de modo a atingir objetivos globais que não seriam viáveis de forma isolada. Além disso, os

*Os autores gostariam de agradecer pelo apoio financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Código de Financiamento 001, do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ).

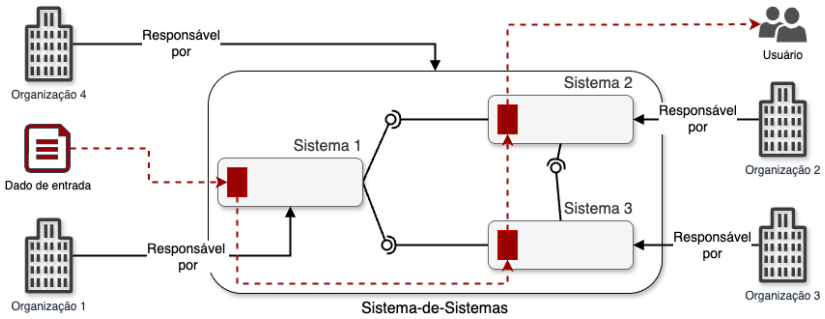


Figura 1. Rastreabilidade dos dados dentro de um SoS.

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 and Sheth 2018]. Em um SoSs, 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 et al. 2022, Curry and Sheth 2018, Lis and Otto 2020]. Entretanto, a literatura aponta a necessidade de mecanismos para o controle do ciclo de vida dos dados dentro dos SoS [Curry and 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. 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. 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 et al. 2017] se mostram como uma solução natural para representar o caminho de derivação dos dados em SoSs [Gammack et al. 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 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 et al. 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 and Moreau 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.

Este artigo está estruturado da seguinte forma. A Seção 2 apresenta o referencial teórico. A Seção 3 revisa trabalhos relacionados ao domínio do governança de dados de SoS. A Seção 4 detalha a abordagem proposta, enquanto a Seção 5 avalia o uso do `PROVGov-SoS` num SoS real. Por fim, a Seção 6 sintetiza os resultados obtidos e discute trabalhos futuros.

2. Uma Introdução aos Dados de Proveniência

Os dados de proveniência são importantes para a governança por fornecer um registro detalhado sobre a origem dos dados e as transformações pelas quais passaram, garantindo transparência, confiabilidade e segurança [Simmhan et al. 2005]. Em diversos domínios, como a biologia, a medicina e a engenharia, o rastreio dos dados de proveniência é essencial para a proteção de direitos, conformidade regulatória, gestão de dados, atribuição de responsabilidades e autenticação da informação [Hasan et al. 2009]. A captura dos dados de proveniência em um formato padronizado e com estrutura processável por máquina torna-se especialmente importante para facilitar a interoperabilidade e o reúso [Magagna et al. 2020]. Dados de proveniência são fundamentais para fornecer serviços confiáveis e seguros, especialmente em ambientes descentralizados nos quais os dados são frequentemente atualizados [Zhao et al. 2009], cenário comum no contexto de SoSs.

O padrão PROV constitui a recomendação do 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 and Miles 2013]. No centro desse conjunto está o PROV-DM [Moreau and 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 esses elementos. 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. As relações entre esses elementos, *e.g.*, *wasGeneratedBy*, *used*, *wasDerivedFrom*, *wasAssociatedWith*, *wasInformedBy*, *actedOnBehalfOf* e *wasAttributedTo*, compõem os relacionamentos entre os elementos citados anteriormente.

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 artigo como formato de representação de dados de proveniência devido ao seu bom desempenho, interoperabilidade, legibilidade e compatibilidade. Embora o PROV-JSON represente um avanço na serialização, ele apresenta limitações quando se trata de consultas, especialmente na ausência de suporte 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. No contexto da abordagem proposta neste artigo, os dados de proveniência foram carregados no Neo4j, um sistema de banco de dados orientado a grafos. Essa escolha se justifica pelo fato de que bancos de dados orientados a grafos oferecem uma correspondência natural com a estrutura do modelo PROV, que é essencialmente ba-

seada em entidades, atividades, agentes e relações representadas como um grafo. Assim, não há 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].

3. Trabalhos Relacionados

A literatura reflete a importância do estudo da proveniência em diferentes áreas do conhecimento [de Oliveira et al. 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. 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 oferece várias possibilidades de visualização que permitem aos usuários monitorar efetivamente o status dos SoS e detectar violações por meio do *framework* REMINDS [Vierhauser et al. 2016]. 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. 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. Ao empregar métricas específicas para avaliar o desempenho dos sistemas constituintes, o *framework* proposto visa aumentar a eficiência operacional e a adaptabilidade diante de condições dinâmicas.

Por sua vez, Kong et al. (2020) apresentam uma abordagem que permite observar o comportamento do SoS sem alterar o código-fonte. 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 abordagem proposta neste artigo distingue-se por tratar especificamente da governança de dados em SoS, indo além do tradicional monitoramento do estado do sistema. Para isso, a solução `PROVGov-SoS` fundamenta-se na captura, persistência e consulta de dados de proveniência estruturados segundo o padrão W3C PROV. Em contraste com os trabalhos anteriores, que concentram seus esforços na otimização de desempenho, na análise de execução ou no monitoramento de requisitos operacionais, a `PROVGov-SoS` coloca ênfase na rastreabilidade de dados. A abordagem registra o 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, bem como para a definição e aplicação de políticas de governança de dados no contexto de SoS.

4. A Abordagem `PROVGov-SoS`

Neste artigo, propomos o uso de dados de proveniência como espinha dorsal para a governança de dados em SoSs, com o objetivo de assegurar que tais dados sejam rastreáveis,

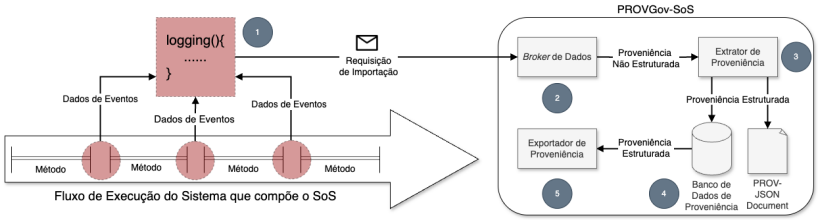


Figura 2. A Arquitetura da Abordagem PROVGov-SoS.

confiáveis e auditáveis [Moreau 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. A Figura 2 ilustra a arquitetura da solução proposta, composta por seis componentes principais: o sistema participante do SoS; o *Broker* de Dados; o Extrator de Proveniência; o repositório de documentos em formato PROV-JSON; o banco de dados de proveniência; e o exportador de proveniência. A seguir, cada um desses componentes é descrito em detalhes.

A execução da PROVGov-SoS tem início em cada um dos sistemas que compõem o SoS. Por meio do uso de Programação Orientada a Aspectos [Kiczales et al. 2001], são interceptados os métodos responsáveis por realizar transformações em dados (passo 1 na Figura 2). A cada invocação desses métodos, informações da transformação realizada e dos dados transformados são capturados automaticamente, incluindo carimbos de data, hora e um identificador do sistema participante que executou o método. Esses dados são então encapsulados em uma mensagem e encaminhados 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 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). Esse componente tem como função processar os dados dos eventos relacionados às transformações realizadas nos sistemas que compõem o SoS, estruturando-os em um grafo de proveniência unificado, conforme as recomendações do padrão PROV. É importante destacar que a construção desse grafo de proveniência ocorre de forma incremental. Embora cada sistema participante mantenha seu próprio grafo de proveniência "local", que representa apenas as transformações executadas internamente, a abordagem PROVGov-SoS visa consolidar essas informações em um grafo de proveniência global do SoS. Esse grafo unificado permite representar as relações de dependência e encadeamento entre os dados ao longo de múltiplos sistemas, viabilizando uma visão completa e integrada da trajetória dos dados dentro do ecossistema.

Para construir o grafo de proveniência global do SoS, a PROVGov-SoS adota o conceito de *bundle*, conforme definido na recomendação PROV. No contexto do PROV, um *bundle* é descrito como a "proveniência da proveniência" [Moreau and Missier 2013], sendo tratado como uma entidade especial que encapsula um subgrafo de proveniência. Neste trabalho, um *bundle* é interpretado como uma transação específica realizada dentro do SoS. Cada *bundle* contém as operações de transformação de dados ocorridas durante a respectiva transação que pode envolver vários sistemas dentro do SoS, proporcionando a granulari-

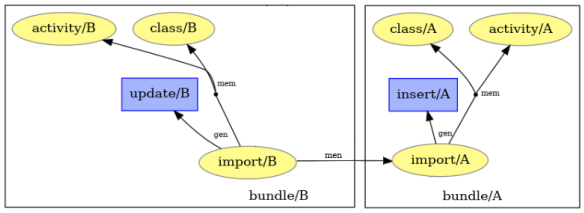


Figura 3. Exemplo de dois *bundles* e as ligações de suas entidades usando o relacionamento *mentionOf* (*men*).

dade 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. Assim, 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 dois *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` (simplificado como *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`. As conexões entre entidades pertencentes a diferentes *bundles* são estabelecidas por meio do relacionamento chamado `MentionOf` [Moreau and 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.

Uma das contribuições deste trabalho reside na 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. Em nossa metodologia, 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 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) *used*, que relaciona uma atividade à entidade utilizada durante sua execução, evidenciando sua dependência operacional; (iii) *wasInvalidatedBy*, que indica o momento em que uma entidade foi invalidada,

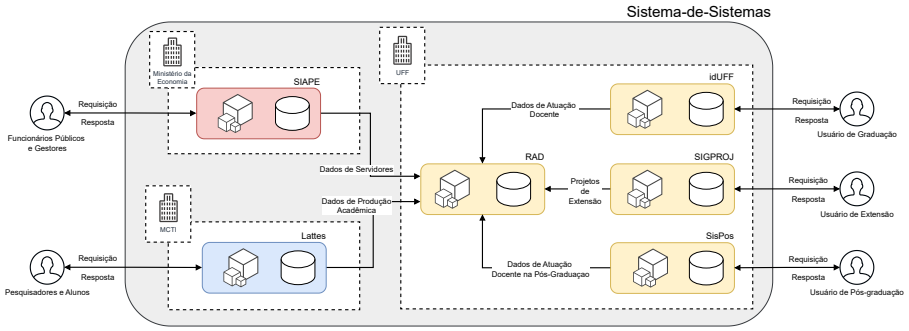


Figura 4. Arquitetura do SoS escolhido para estudo de viabilidade do PROVGov-SoS

destruída ou expirada, como resultado de uma atividade específica; (iv) *hadMember*, utilizado para representar que uma entidade do tipo coleção contém outras entidades como seus membros; e (v) *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.

Por fim, 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). Na implementação atual da PROVGov-SoS, utilizamos o Neo4J [Neo4j 2025] como sistema de banco de dados orientado a grafos. 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 et al. 2017]. O código-fonte da versão atual da PROVGov-SoS pode ser obtido no repositório do GitHub em https://github.com/dew-uff/PROVGov-SoS.

5. Avaliação da PROVGov-SoS em um SoS Real

Para avaliar a abordagem PROVGov-SoS, realizamos um estudo de viabilidade utilizando um SoS real, desenvolvido pela administração central da Universidade Federal Fluminense. O sistema denominado Relatório de Atividades (RAD)[1] 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.

Os seis sistemas que compõem o SoS são ilustrados na Figura 4 e descritos a seguir: (i) o idUFF, 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, sistema de informação mantido pelo Conselho Nacional de Desenvolvimento

[1]https://app.uff.br/rad/

Científico e Tecnológico (CNPq), que agrega currículos de pesquisadores e estudantes; (iii) o SIAPE, sistema do governo federal brasileiro utilizado para gerenciar os registros funcionais dos servidores públicos civis; (iv) o SIGPROJ, plataforma destinada ao registro, acompanhamento e avaliação de projetos de extensão em universidades brasileiras; e (v) o SisPOS, 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 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.

A seguir, descrevemos passo a passo a condução do estudo de viabilidade da `PROVGov-SoS` com o objetivo de apoiar a governança de dados no SoS RAD. O primeiro 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: (i) Conjuntos de Dados manipulados: representando coleções que agregam todas as tuplas consumidas ou geradas por uma atividade; (ii) Produtos: entidades que estão associadas a um Conjunto de Dados e contêm atributos específicos, *e.g.*, um artigo publicado; e (iii) Propriedades: entidades que fornecem informações detalhadas, como a 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 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.

Conforme mencionado anteriormente, a captura dos dados de proveniência nos sistemas que compõem o SoS foi orientada à aplicação [Singh et al. 2019] 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. Para os sistemas externos, foi desenvolvido um submódulo em Java, seguindo o padrão de projeto *Facade* [Gamma et al. 1995], que encapsula o sistema externo e fornece uma interface simplificada de acesso. Foi nessa interface que os pontos de captura de proveniência foram inseridos, garantindo que todas as importações fossem devidamente registradas e processadas. Na versão atual, a `PROVGov-SoS` utiliza a biblioteca ProvToolBox, versão 2.0.4, que permite a criação de descrições baseadas no PROV e a conversão entre diferentes formatos, como RDF, PROV-XML, PROV-N e PROV-JSON. A biblioteca AspectJ, uma extensão para programação orientada a aspectos, foi empregada para detectar e reagir a comportamentos específicos durante a execução dos sistemas do SoS.

Neste estudo de viabilidade, foram analisados dados processados ao longo do ano de 2023. Todas as produções acadêmicas, aulas de graduação e pós-graduação, bem como dados pessoais, foram importados simulando o uso real do SoS. A partir disso, foram realizadas consultas à base de dados de proveniência da `PROVGov-SoS` com o objetivo de identificar e compreender problemas de importação frequentemente relatados pelos usuários finais. Um dos casos mais citados envolvia situações em que, após determinadas importações de dados para o SoS, a carga horária de uma turma (*ch*) era registrada como zero, o que constitui um erro para algumas disciplinas. Para investigar a causa desse problema, utilizamos o banco de dados de proveniência no Neo4j (versão 5.26.0). Todas as consultas foram elaboradas em linguagem Cypher, sem o uso do pacote APOC.

Inicialmente, foi realizada uma consulta 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, apresentada na Figura 5(a), retorna esses nós juntamente com seus relacionamentos, permitindo isolar os dados que representam o estado final do sistema, sem qualquer histórico de atualização ou correção posterior, tanto em formato visual (Figura 5(b)) quanto em formato JSON (Figura 5(c)).

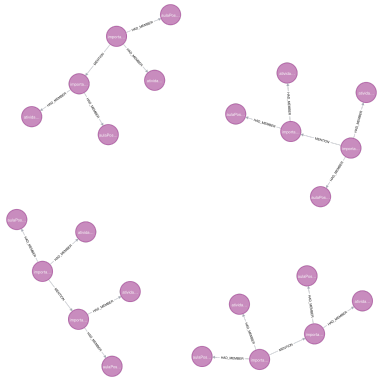
Em seguida, foi implementada uma consulta para identificar os nomes das disciplinas vinculadas às entidades cujo campo de carga horária se encontra vazio, bem como contabilizar a frequência de ocorrência de cada uma delas. O objetivo dessa consulta é identificar quais disciplinas são mais afetadas por essa inconsistência durante uma importação anual de dados. A Figura 6(a) apresenta a consulta em linguagem Cypher, enquanto a Figura 6(b) mostra um fragmento do resultado correspondente em formato JSON.

Finalmente, a terceira consulta 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 Figura 7(a) apresenta a consulta correspon-

a

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

b



c


```
[
 {
  "n": {
   "identity": 1,
   "labels": [
    "Entity"
   ],
   "properties": {
    "ch": "0",
    "ignoredFields": "null",
    "localidade": "[Anonimizado]",
    "name": "aulaPosGraduacao/1/17446802",
    "disciplinaNome": "BIOESTATÍSTICA I",
    "turmaCodigo": "TM04220221012",
    "qtdAlunos": "7",
    "disciplinaCodigo": "CMM10754",
    "idAtividade": "1",
    "id": "1"
   },
   "elementId": "4:e9180ad6-57e9-4dad-85b7:1"
  },
```


Figura 5. Consulta Q1 - entidades que não sofreram alterações posteriores (a) Consulta em Cypher, (b) Grafo resultante, (c) Trecho do resultado em JSON.

a

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

b


```
[
 {
  "disciplina": "DISSERTAÇÃO",
  "ocorrencias": 187
 },
 {
  "disciplina": "ESTÁGIO DOCÊNCIA",
  "ocorrencias": 93
 },
 {
  "disciplina": "TESE",
  "ocorrencias": 83
 },
```


Figura 6. Consulta Q2 - disciplinas vinculadas a entidades com carga horária vazia (a) Consulta em Cypher, (b) Trecho do resultado em JSON.

dente, redigida em linguagem Cypher, enquanto a Figura 7(b) ilustra um trecho do resultado obtido, representado no formato JSON.

O experimento 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 et al. 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. Conclusões e Trabalhos Futuros

Este artigo 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 refe-

a

```
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
```

b


```
[
 {
  "n.name":
"aulaPosGraduacao/4476/1744893030548",
  "m.name":
"aulaPosGraduacao/4476/1744893030541",
  "diffs": [
   "ch",
   "qtdAlunos"
  ]
 },
 {
  "n.name":
"aulaPosGraduacao/5015/1744893055406",
  "m.name":
"aulaPosGraduacao/5015/1744893055415",
  "diffs": [
   "ch",
   "qtdAlunos"
  ]
 },
```


Figura 7. Consulta Q3 - versões de entidades que sofreram alterações posteriores a carga com problemas (a) Consulta em Cypher, (b) Trecho do resultado em JSON.

rência 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 et al. 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 por uma universidade federal brasileira. 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 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, bem investigar técnicas de coleta menos intrusivas e mais eficientes em termos de desempenho. Por fim, vislumbra-se também o desenvolvimento de mecanismos analíticos para a detecção automatizada de anomalias e apoio à tomada de decisão, além da aplicação da `PROVGovSoS` em domínios diferentes.

Referências

Allen, M. D., Chapman, A., Seligman, L., and Blaustein, B. (2011). Provenance for collaboration: Detecting suspicious behaviors and assessing trust in information. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 342–351. IEEE.

Almeida, R., Silva, W. M. C. D., Castro, K., Araújo, A. P. F. D., Walter, M. E. M. T., Lifschitz, S., and Holanda, M. (2019). Managing data provenance for bioinformatics workflows using aprovbio. *International Journal of Computational Biology and Drug Design*, 12(2):153–170.

Anuyah, S., Bolade, V., and Agbaakin, O. (2024). Understanding graph databases: a comprehensive tutorial and survey. *arXiv preprint arXiv:2411.09999*.

Calabro, A., Daoudagh, S., Marchetti, E., Mayo, F., Marchiori, M., and Filipe, J. (2021). Mentors: Monitoring environment for system of systems. In *WEBIST*, pages 291–298.

Cavalcante, E., Batista, T., and Oquendo, F. (2024). Looking back and forward: A retrospective and future directions on software engineering for systems-of-systems. *Journal of Software: Evolution and Process*, 36(10):e2697.

Chreim, A., Yiwen, C., Smahi, A., Jiang, J., and Merzouki, R. (2024). Towards supervision of stochastic system of systems engineering: A multi-level hypergraph approach. *IEEE Access*.

Curry, E., Scerri, S., and Tuikka, T. (2022). *Data spaces: design, deployment and future directions*. Springer Nature.

Curry, E. and Sheth, A. (2018). Next-generation smart environments: From system of systems to data ecosystems. *IEEE Intelligent Systems*, 33(3):69–76.

de Oliveira, W. M., de Oliveira, D., and Braganholo, V. (2018). Provenance analytics for workflow-based computational experiments: A survey. *ACM Comput. Surv.*, 51(3):1–25.

Fu, X., Wojak, A., Neagu, D., Ridley, M., and Travis, K. (2011). Data governance in predictive toxicology: A review. *Journal of cheminformatics*, 3:1–16.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., USA.

Gammack, D., Scott, S., and Chapman, A. P. (2016). Modelling provenance collection points and their impact on provenance graphs. In *International Provenance and Annotation Workshop (IPAW)*, pages 146–157. Springer.

Gil, Y. and Miles, S. (2013). Prov model primer. https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/.

Groth, P. and Moreau, L. (2013). Prov-overview. https://www.w3.org/submissions/2013/SUBM-prov-json-20130424/.

Hasan, R., Sion, R., and Winslett, M. (2009). Preventing history forgery with secure provenance. *ACM Transactions on Storage (TOS)*, 5(4):1–43.

Herschel, M., Diestelkämper, R., and Ben Lahmar, H. (2017). A survey on provenance: What for? what form? what from? *VLDB J.*, 26(6):881–906.

Huynh, T. D., Jewell, M. O., Keshavarz, A. S., Michaelides, D. T., Yang, H., and Moreaun, L. (2013). Prov-json serialization. https://www.w3.org/TR/prov-overview/.

Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., and Griswold, W. G. (2001). An overview of aspectj. In *European Conference on Object-Oriented Programming (ECOOP)*, pages 327–354. Springer.

Kong, S., Lu, M., Li, L., and Gao, L. (2020). Runtime monitoring of software execution trace: Method and tools. *IEEE Access*, 8:114020–114036.

Kritzinger, L. M., Krismayer, T., Rabiser, R., and Grünbacher, P. (2019). A user study on the usefulness of visualization support for requirements monitoring. In *Working Conference on Software Visualization (VISSOFT)*, pages 56–66. IEEE.

Lis, D. and Otto, B. (2020). Data governance in data ecosystems–insights from organizations. In *Americas Conference on Information Systems (AMCIS)*.

Magagna, B., Goldfarb, D., Martin, P., Atkinson, M., Koulouzis, S., and Zhao, Z. (2020). Data provenance. In *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences: A Reference Model Guided Approach for Common Challenges*, pages 208–225. Springer.

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering: The Journal of the International Council on Systems Engineering*, 1(4):267–284.

Moreau, L. (2013). Provtoolbox. java library to create and convert w3c prov data model representations. https://lucmoreau.github.io/ProvToolbox/.

Moreau, L., Batlajery, B. V., Huynh, T. D., Michaelides, D., and Packer, H. (2017). A templating system to generate provenance. *IEEE Transactions on Software Engineering*, 44(2):103–121.

Moreau, L. and Lebo, T. (2013). Prov-links. https://www.w3.org/TR/2013/NOTE-prov-links-20130430/.

Moreau, L. and Missier, P. (2013). *PROV-DM: The PROV Data Model*. World Wide Web Consortium, W3C.

Neo4j (2025). Neo4j graph database. https://neo4j.com/product/neo4j-graph-database. Accessado em 12 Mar. 2025.

Simmhan, Y. L., Plale, B., and Gannon, D. (2005). A survey of data provenance in e-science. *SIGMOD Record*, 34(3):31–36.

Singh, J., Cobbe, J., and Norval, C. (2019). Decision provenance: Harnessing data flow for accountable systems. ieee access 7 (2019), 6562–6574.

Vierhauser, M., Rabiser, R., Grünbacher, P., Seyerlehner, K., Wallner, S., and Zeisel, H. (2016). Reminds: A flexible runtime monitoring framework for systems of systems. *Journal of Systems and Software*, 112:123–136.

Wercelens, P., da Silva, W., Hondo, F., Castro, K., Walter, M. E., Araújo, A., Lifschitz, S., and Holanda, M. (2019). Bioinformatics workflows with nosql database in cloud computing. *Evolutionary Bioinformatics*, 15:1176934319889974.

Zhao, J., Miles, A., Klyne, G., and Shotton, D. (2009). Linked data and provenance in biological data webs. *Briefings in bioinformatics*, 10(2):139–152.