

5LTEP-L4: An Open-Source CKAN Toolkit for Provenance-Enabled Observability of Open Government Data

Luciano Santos Pinheiro
lsp3@cesar.school
CESAR School
Recife, PE, Brazil

Anderson Tenório Sérgio
ats@cesar.school
CESAR School
Recife, PE, Brazil

ABSTRACT

We present 5LTEP-L4, an open-source Python toolkit that turns any CKAN-based Open Government Data (OGD) portal into a source of W3C PROV-DM provenance records. Through the CKAN Web API, it fingerprints dataset metadata with SHA-256, classifies each change as a content modification, schema drift, retroactive alteration, or clean update, and publishes PROV-O JSON-LD records in an append-only Git audit trail, orchestrated by GitHub Actions with no dedicated hosting. A 109-day deployment on the IBAMA portal (430 scheduled cycles, 97.2% successful, 79 datasets) detected 31 changes, all confirmed against an independent snapshot diff; controlled fault injection reached 632/632 correct classifications. The toolkit ran unmodified on two further CKAN portals (Recife, 223 datasets; ANEEL, 72), and its records load in the prov reference library. Code is available under the MIT license at <https://github.com/lsp3cesarschool/5ltep-layer4>.

KEYWORDS

data provenance, open government data, W3C PROV, linked data, CKAN, observability, GitHub Actions

1 INTRODUCTION

OGD portals are key Web infrastructure for democratic transparency, evidence-based policy, and machine learning research [2]. In Brazil, publication is mandated by the Lei de Acesso à Informação (LAI, Law 12.527/2011) [12] and Decree 8.777/2016 [13]; as of July 2026, 19,379 datasets were listed on the federal portal (dados.gov.br), and many agencies and cities publish on CKAN, whose Action API exposes dataset metadata over HTTP. Yet an accountability gap persists: most portals keep no machine-readable provenance, so when a dataset is modified or silently altered, no standard audit trail records *what* changed, *when*, *how*, or under *whose* custody; OGD quality studies already report the lack of data-source traceability as a recurring problem [19]. This undermines trust, reproducibility, and compliance and, for ML practitioners, lets undetected data drift silently degrade model performance [5, 14, 16].

The Five-Layer Trust Engineering Pyramid (5L-TEP) [11] places observability [8] and provenance (Layer 4) at the core of OGD trust, but offered no implementation. 5LTEP-L4 fills this gap as a *Web data tool*: it consumes a Web API, emits Linked Data in W3C vocabularies (PROV-O over JSON-LD [7, 9]), and targets datasets, matching the WFA call for applications that bring new analysis methods

Table 1: Comparison with related tools.

Tool	CKAN harvest	Change classes	PROV output	Tamper-evident	No hosting
CKAN activity stream	✓	–	–	–	~
Portal Watch [10]	✓	~	–	–	–
Deequ [15]	–	~	–	–	–
prov library [6]	–	–	✓	–	n/a
5LTEP-L4	✓	✓	✓	✓	✓

to datasets. Its contributions are: (i) an operational change taxonomy for CKAN metadata; (ii) a PROV-DM mapping with content-addressed entities and a dual-agent model; (iii) a zero-hosting deployment pattern; and (iv) an evaluation covering long-term operation, classification accuracy, portability, and interoperability.

2 RELATED TOOLS

Provenance has long been studied for databases [3] and for computational tasks and workflows [4, 17]; here we focus on tools for monitoring OGD portals. Table 1 positions 5LTEP-L4 (✓ = yes, ~ = partial, – = no). Portal Watch [10, 18] crawls hundreds of portals to assess metadata quality and its evolution, but runs on dedicated infrastructure and emits no provenance graph. Deequ [15] verifies data-quality constraints on Spark tables, not portal metadata. CKAN’s activity stream logs user actions inside the portal database, without content-addressed identifiers or standard vocabulary, and is controlled by the same custodian whose edits it records. The prov library [6] is the reference Python implementation of PROV-DM: it models provenance, serialises it to PROV-N, PROV-JSON, PROV-XML, and PROV-O, and plots it as graphs, but does not harvest portals or detect and classify changes. The tools are complementary; 5LTEP-L4 supplies the observation layer that produces PROV documents, and emits PROV-O JSON-LD directly to keep a single runtime dependency (requests). We verified interoperability by loading its records with prov (Section 4).

3 THE 5LTEP-L4 TOOLKIT

3.1 Objectives and Architecture

The toolkit aims to (1) emit a PROV-DM record for every detected change in a CKAN portal; (2) identify each dataset state by a content hash and link it to its predecessor; (3) keep per-dataset, append-only audit trails; and (4) run on free CI infrastructure. Figure 1 shows the pipeline: three stateless Python modules run every six hours on GitHub Actions. The *Harvester* calls `package_list/package_show` with exponential back-off (3 attempts, base 2 s); the *Hash Engine*

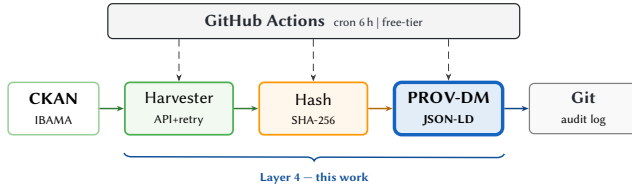


Figure 1: 5LTEP-L4 pipeline. GitHub Actions orchestrates three modules: Harvester, Hash Engine, and PROV-DM Mapper (highlighted), which implement Layer 4 of 5L-TEP. Records are stored immutably in Git.

fingerprints and classifies; the *PROV-DM Mapper* writes JSON-LD records that the workflow commits to Git.

3.2 Main Functionalities

Change detection. For each dataset, the Hash Engine computes h_t , a SHA-256 over the canonical JSON (sorted keys) of stable metadata fields (title, notes, metadata_modified, and each resource’s name, format, URL, size, and last modification), excluding volatile API fields, plus a *manifest hash* over the resource names and formats. For multi-resource datasets (e.g., CSV, JSON, and PDF files), a change in any single resource propagates to the dataset-level event. Comparing with the stored state yields: CLEAN_UPDATE ($h_t = h_{t-1}$ or first observation; INFO); SCHEMA_DRIFT (manifest changed; CRITICAL); RETRO_ALTER ($h_t \neq h_{t-1}$ without metadata_modified advancing, i.e., an undocumented edit; CRITICAL); and CONTENT_MOD (change with an advancing timestamp; WARNING).

PROV-DM mapping. Each non-clean event becomes a JSON-LD document in which: the dataset state is a prov:Entity whose IRI is the portal URL, dataset id, and a hash fragment, so any change yields a new, immutable entity linked to the previous one by prov:wasDerivedFrom; the monitoring run is a prov:Activity with ISO-8601 timestamps that prov:used the API endpoint; and a dual-agent model records both the GitHub Actions runner (prov:SoftwareAgent, at a given commit) and the originating government agency (prov:wasAttributedTo), separating observer from data custodian [17]. Entities also carry the change type and severity. Listing 1 shows an abridged record from the IBAMA deployment.

```

{"@context": {"prov": "http://www.w3.org/ns/prov#",
  "5lstep": "...", "portal": "https://dadosabertos.ibama.gov.br/"},
"@graph": [
  {"@id": "portal:dataset/4b292c2e-...#87db6229b63c",
  "@type": ["prov:Entity", "5lstep:DatasetSnapshot"],
  "prov:wasGeneratedBy": {"@id": "5lstep:run-20260922T162835Z"},
  "prov:wasDerivedFrom": {"@id": "portal:dataset/4b292c2e-...#df7bf245d8ff"},
  "prov:wasAttributedTo": {"@id": "portal:organization/ibama"},
  "5lstep:changeType": "CONTENT_MOD", "5lstep:severity": "WARNING"},
  {"@id": "5lstep:run-20260922T162835Z", "@type": "prov:Activity",
  "prov:used": {"@id": "portal:api/3/action/package_show?id=4b292c2e-..."},
  "prov:wasAssociatedWith": {"@id": "https://github.com/.../5lstep-layer4@
  ..."}],
  ... ]}
  
```

Listing 1: Abridged PROV-DM record (JSON-LD, IRIs compacted with the portal prefix) for a file re-upload detected in an IBAMA dataset.

Append-only audit storage. Records are appended to one JSON-LD file per dataset in a Git repository; Git commit hashes provide tamper-evidence without a dedicated blockchain. The JSON-LD format enables direct ingestion into SPARQL endpoints for lineage queries (e.g., “which datasets had retroactive alterations in the last 30 days?”). Since v1.0.2, each record also states how the dataset changed (changed fields, relocated resource URLs, and zip/plain packaging switches), and every cycle rebuilds changes.md, a human-readable log of what changed, when, and how, linked from the README.

GitHub Actions orchestration. The monitoring workflow runs six steps every six hours: (1) checkout; (2) Python 3.11 setup; (3) dependency installation; (4) main.py, which writes critical-change signals to GITHUB_OUTPUT; (5) an automated commit of the provenance logs; and (6) a conditional alert step that fails the workflow on SCHEMA_DRIFT or RETRO_ALTER. Because the commit comes first, the maintainer is alerted without losing the record. An independent cross-check workflow compares the latest snapshot with the live portal daily (03:30 UTC) and fails when coverage drops below 90% or divergences remain unreconciled for more than 7 h. Each failed run triggers GitHub’s built-in e-mail notification, so alerts reach the maintainer at no cost, without a mail server or credentials; during the deployment, each of the 19 failed runs (12 monitoring, 7 cross-check) produced such an e-mail. A weekly workflow (Sundays, 03:00 UTC) gzip-compresses snapshots older than 90 days.

FAIR and DQV alignment. Records support the FAIR principles [20]: *Findable* via content-addressable IRIs, *Interoperable* via W3C vocabularies, and *Reusable* via documented lineage. Linked Data quality assessment also relies on provenance information to measure verifiability and trustworthiness [21]. The records extend naturally to the W3C Data Quality Vocabulary [1], so quality metrics can be published alongside provenance.

Adapting to other portals. Monitoring a new CKAN portal needs no code change: the portal is set by the CKAN_PORTAL_URL variable (or --portal), with optional CKAN_ORG_FILTER and MAX_DATASETS. Adopters fork the repository, set one repository variable, and enable Actions; the monitoring and cross-check workflows both read it. Only read access to the public API is required.

License and availability. 5LSTEP-L4 is released under the MIT license (permissive, without copyleft), which allows academic, governmental, and commercial use, at <https://github.com/lsp3cesarschool/5lstep-layer4>. The repository includes the Python source (src/), the GitHub Actions workflows (.github/workflows/), 38 automated tests, the live provenance logs (provenance_logs/), the scripts that reproduce the evaluation below (evaluation/), and a README with installation and configuration instructions. The only runtime dependency beyond the Python standard library (hashlib, json) is requests.

4 DEMONSTRATION AND EVALUATION

Long-running deployment. We deployed 5LSTEP-L4 on the CKAN portal of IBAMA, Brazil’s federal environmental agency (<https://dadosabertos.ibama.gov.br>), whose datasets cover environmental

Table 2: Fault injection over the 79 IBAMA datasets.

Injected change (per dataset)	Expected class	Correct
None (control)	CLEAN_UPDATE	79/79
Notes edit; file re-upload	CONTENT_MOD	158/158
Resource added; removed; format	SCHEMA_DRIFT	237/237
Same as row 2, no timestamp	RETRO_ALTER	158/158
Tags/license only (unhashed)	none	0/79 found

enforcement, licensing, and biodiversity, and monitored it from 9 June to 27 September 2026 (109 days) without human intervention. Of 430 scheduled cycles, 418 succeeded (97.2%); the 12 failures aborted inside the monitoring step and later cycles recovered automatically, with a longest gap of 24 h between successful cycles. The portal grew from 71 to 79 datasets, and each addition was recorded as a baseline. Content addressing stored 30 unique snapshots for 419 recorded cycles (92.8% less storage than archiving every cycle). The independent cross-check reported IN_SYNC for 100% of the datasets (79/79), last verified on 27 September 2026 at 09:27 UTC.

Detected events. The toolkit recorded 31 events, all CONTENT_MOD, in 22 of the 79 datasets (28%); one dataset changed 8 times. Field-level analysis shows that 29 events reflect IBAMA’s migration of file downloads to cloud storage: 105 resource URLs changed (69 moved to a new host and 54 switched between zip and plain files), while the declared formats stayed the same, so consumer scripts would break silently. Two events were description edits (one together with a URL change), one changed only non-fingerprinted resource descriptions (detected through `metadata_modified`), and 7 also changed the license title, a field outside the fingerprint. No SCHEMA_DRIFT or RETRO_ALTER occurred: CKAN refreshes `metadata_modified` on every API edit, so retroactive alterations can only come from out-of-band edits, which makes their absence plausible for a well-run portal.

Classification accuracy. As ground truth, we independently diffed every pair of consecutive stored snapshots. This yielded exactly 31 dataset modifications (all with an advancing timestamp and an unchanged manifest) plus the 8 additions. The toolkit’s 31 labels match them one-to-one, with no false positives or misses at snapshot granularity. Because real critical events were absent, we also injected controlled changes into all 79 datasets of the latest snapshot (Table 2). All 632 in-scope cases were classified correctly; edits confined to tags or license (79 cases) were not detected, a limitation of the fingerprint’s scope rather than of the classifier.

Portability. Without any code change, we ran three cycles against two other CKAN portals: the Recife city portal (223 datasets, CKAN 2.11.5) and the federal electricity regulator ANEEL (72 datasets). All datasets were harvested without errors, in about 60 s and 18 s per cycle (≈ 0.26 s per dataset), and deduplication stored a single snapshot per portal, since no changes occurred during the test.

Interoperability. Loading all 31 IBAMA records with prov 3.2.2 initially failed: its PROV-O reader requires every IRI to be expressible as a qualified name, and our records did not declare the portal namespace. Release v1.0.1 adds a portal prefix to the JSON-LD context and makes the software-agent IRI absolute. All 31 records

Table 3: Projected monthly GitHub Actions usage.

Monitoring frequency	Runs/month	Billed min	% of free tier
Every 6 h (deployed)	120	298	14.9%
Every 12 h	60	184	9.2%
Daily	30	127	6.4%

then load (31 entities, 31 activities, 62 agents, 31 derivations), and a regression test now checks this in CI.

Resource consumption. A monitoring cycle takes a median of 36 s (p90 141 s). In the last 30 days, the three workflows used 290 billed minutes, about 14.5% of GitHub’s 2,000-minute free tier for private repositories (public repositories are not charged). Table 3 projects this measured usage (1.9 billed minutes per monitoring run, plus 70 minutes per month for the cross-check and compression workflows) to other monitoring frequencies; it assumes all 120 scheduled runs per month, whereas delayed cron jobs executed 116 in the measured period. Since harvesting takes about 0.26 s per dataset, a portal with 2,000 datasets would need roughly 9 min per cycle, which is still within the free tier at the 6-hour frequency (about 60%).

5 DEVELOPMENT CHALLENGES

Deployment exposed issues that are invisible in local tests: (i) the cross-check job ran `git pull --rebase` on a dirty tree and raced with the monitor’s commits, failing four consecutive days, which we fixed by committing before rebasing with `--autostash`; (ii) early workflows turned transient portal outages into failure e-mails, so alerts now fire only on CRITICAL events or a stale cross-check, after the commit; (iii) GitHub’s scheduler delays cron jobs, which motivated a 7-hour staleness tolerance; (iv) storing a full snapshot every cycle grew the repository quickly, which led to content-addressed deduplication and gzip compression; (v) TLS inspection by local antivirus broke certificate validation when testing new portals; and (vi) token and context limits of the AI assistants used during development pushed us toward small, individually tested modules.

6 PERSPECTIVES AND LIMITATIONS

Academic. As a research prototype, 5LTEP-L4 shows that standards-compliant provenance monitoring of OGD portals is feasible at zero cost. It supports LAI transparency audits and CGU accountability reporting in Brazil’s data ecosystem, reproducible studies of OGD evolution, and drift detection on OGD training data [16].

Social. Federal, state, and municipal CKAN portals, such as those of Recife and ANEEL, can adopt the toolkit with read-only access to the standard API, which gives citizens and auditors an independent, verifiable record of how public datasets change. Records also respect privacy (LGPD, Law 13.709/2018), since only publicly available metadata is monitored and file contents are not stored.

Entrepreneurial. The MIT license allows integration into open-data platforms, civic-tech products, and public-administration toolchains, and the zero-hosting model, with e-mail alerts delivered by GitHub at no cost, lowers the barrier for municipalities and NGOs with small infrastructure budgets.

Limitations and future work. Detection is limited to metadata, so file content replaced in place and edits to non-fingerprinted fields go unnoticed; changes reverted within one polling interval are invisible; and the portability test covered only a few cycles. Hash collisions are negligible (probability $\approx 10^{-65}$ for 10^6 snapshots). Relocations and packaging changes, as observed at IBAMA, are described in the records but still classified as ordinary content modifications, although they can break consumers. Future work includes multi-portal validation across CKAN configurations, content-level change tracking via resource downloads, a configurable fingerprint with a dedicated class for resource relocation, W3C DQV integration for publishing quality metrics, visual provenance explorers, and integrations with MLOps drift alerts.

7 ETHICAL CONSIDERATIONS

This work involves no human subjects. The toolkit only reads publicly available metadata from governmental CKAN portals and does not collect, store, or process personal data; CNS Resolution 510/2016 therefore does not apply.

ACKNOWLEDGMENTS

The authors declare that generative AI tools (SciSpace and Claude Code) were used to support code, documentation and manuscript drafting, in compliance with the WebMedia 2026 guidelines; all content was reviewed by the authors.

REFERENCES

- [1] Riccardo Albertoni and Antoine Isaac. 2016. Data on the Web Best Practices: Data Quality Vocabulary. W3C Working Group Note. <https://www.w3.org/TR/vocab-dqv/>
- [2] Judie Attard, Fabrizio Orlandi, Simon Scerri, and Sören Auer. 2015. A Systematic Review of Open Government Data Initiatives. *Government Information Quarterly* 32, 4 (2015), 399–418. doi:10.1016/j.giq.2015.07.006
- [3] James Cheney, Laura Chiticariu, and Wang-Chiew Tan. 2009. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases* 1, 4 (2009), 379–474. doi:10.1561/19000000006
- [4] Juliana Freire, David Koop, Emanuele Santos, and Cláudio T. Silva. 2008. Provenance for Computational Tasks: A Survey. *Computing in Science & Engineering* 10, 3 (2008), 11–21. doi:10.1109/MCSE.2008.79
- [5] Bernardo Gallo, Lúcia Drummond, and José Viterbo. 2025. Mitigating Concept Drift in Job Execution Time Prediction Models. In *Proceedings of the 13th Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*. SBC, Fortaleza, CE, Brazil, 81–88. doi:10.5753/kdmile.2025.247724
- [6] Trung Dong Huynh. 2026. prov: A Python Library for the W3C Provenance Data Model (version 3.2.2). Software, MIT License. <https://github.com/trungdong/prov>
- [7] Timothy Lebo, Satya Sahoo, and Deborah McGuinness. 2013. PROV-O: The PROV Ontology. W3C Recommendation. <https://www.w3.org/TR/prov-o/>
- [8] Charity Majors, Liz Fong-Jones, and George Miranda. 2022. *Observability Engineering: Achieving Production Excellence*. O'Reilly Media, Sebastopol, CA.
- [9] Luc Moreau and Paolo Missier. 2013. PROV-DM: The PROV Data Model. W3C Recommendation. <https://www.w3.org/TR/prov-dm/>
- [10] Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. 2016. Automated Quality Assessment of Metadata across Open Data Portals. *Journal of Data and Information Quality* 8, 1, Article 2 (2016), 29 pages. doi:10.1145/2964909
- [11] Luciano Santos Pinheiro, Cristiane de Holanda de Barros e Silva, Vitor Barros Aquino, Thays Maria da Conceição Silva Carvalho, Cristiano Vale do Rego Barros Filho, and Washington Henrique Carvalho Almeida. 2026. Towards Trust Engineering in Open Data Systems: A Layered Conceptual Framework Integrating Quality Assurance and Governance Perspectives. In *SOFTENG 2026: The Twelfth International Conference on Advances and Trends in Software Engineering*. IARIA, Venice, Italy, 21–28. https://www.thinkmind.org/library/SOFTENG/SOFTENG_2026/softeng_2026_1_40_90029.html
- [12] Presidência da República. 2011. Lei nº 12.527, de 18 de novembro de 2011: Lei de Acesso à Informação. Diário Oficial da União. https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/112527.htm
- [13] Presidência da República. 2016. Decreto nº 8.777, de 11 de maio de 2016: Institui a Política de Dados Abertos do Poder Executivo Federal. Diário Oficial da União. https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2016/decreto/d8777.htm
- [14] Fabiana Santos, Lucas Giusti, Diego Carvalho, Eduardo Ogasawara, and Jorge Soares. 2024. Avaliação de Desvios de Conceitos Reais e Virtuais nos Atrasos de Voos em São Paulo nos Períodos Pré, Intra e Pós-Pandemia. In *Proceedings of the 39th Brazilian Symposium on Databases (SBBd)*. SBC, Florianópolis, SC, Brazil, 827–833. doi:10.5753/sbbd.2024.243111
- [15] Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann, and Andreas Grafberger. 2018. Automating Large-Scale Data Quality Verification. *Proceedings of the VLDB Endowment* 11, 12 (2018), 1781–1794. doi:10.14778/3229863.3229867
- [16] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. Hidden Technical Debt in Machine Learning Systems. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*. Curran Associates, Montréal, Canada, 2503–2511.
- [17] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. 2005. A Survey of Data Provenance in e-Science. *ACM SIGMOD Record* 34, 3 (2005), 31–36. doi:10.1145/1084805.1084812
- [18] Jürgen Umbrich, Sebastian Neumaier, and Axel Polleres. 2015. Quality Assessment and Evolution of Open Data Portals. In *Proceedings of the 3rd International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, Rome, Italy, 404–411. doi:10.1109/FiCloud.2015.82
- [19] Antonio Vetrò, Lorenzo Canova, Marco Torchiano, Camilo Orozco Minotas, Raimondo Iemma, and Federico Morando. 2016. Open Data Quality Measurement Framework: Definition and Application to Open Government Data. *Government Information Quarterly* 33, 2 (2016), 325–337. doi:10.1016/j.giq.2016.02.001
- [20] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. 2016. The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Scientific Data* 3 (2016), 160018. doi:10.1038/sdata.2016.18
- [21] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. 2016. Quality Assessment for Linked Data: A Survey. *Semantic Web* 7, 1 (2016), 63–93. doi:10.3233/SW-150175