Systems security assurance as (micro) publishing?

Declarative markup for systems description and assessment

Wendell Piez

Information Technology Laboratory

National Institute of Standards and Technology (USA)

What is this about?

(terms become meaningful only in context)

Systems Security Assurance

Declarative markup

(Micro) publishing

Assessment

Systems Description

… workflow …

Systems security assurance

An industry

A set of activities and commitments

FISMA (Federal Information System Management Act) – *US government agency requirements*

NIST SP800-37: Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy

NIST SP 800-53: Security and Privacy Controls for Federal Information Systems and Organizations

ISO/IEC 27001: INFORMATION SECURITY MANAGEMENT

NIST CSF (Cybersecurity Framework) – *open framework for private and public sectors*

PCI-DSS (Payment Card Industry – Data Security Standard)

HIPAA (Health Insurance Portability and Accountability Act)

CCM (Cloud Security Alliance Cloud Control Matrix)

… and many more (examples reflect US context) …

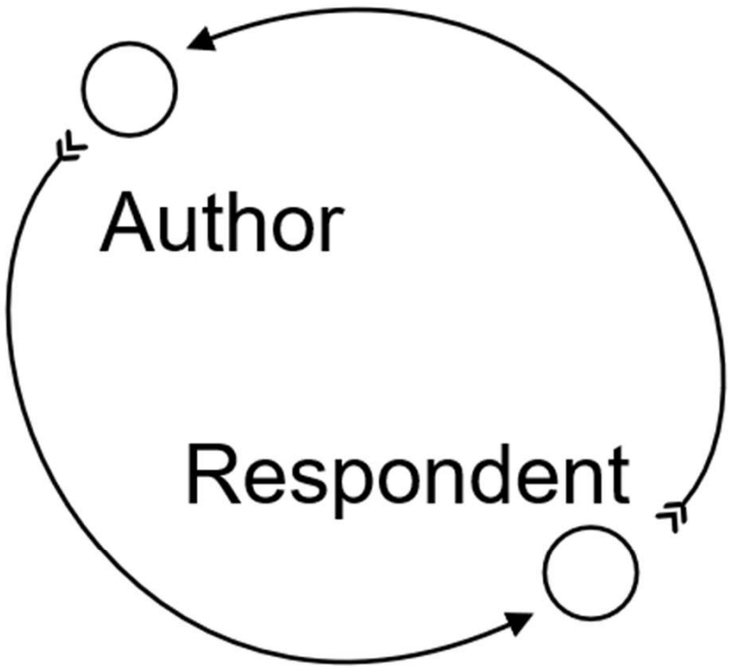
A domain

A set of questions

- **What is security?**
- **What is systems security?**
- **What is assurance?**
- **How do I know I am assured?**
- **How does my partner know I am assured?**
- ***How do I assure my partner?***
  - Documentation, attestation
  - Demonstration
  - Legal requirements

(Micro) publishing

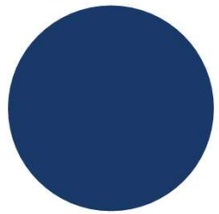
Or: why the “office document” won the format wars



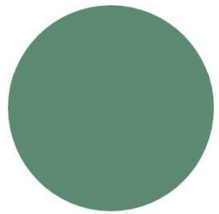
“Publishing” is information capture and refinement

- Creation
- Revision
- Presentation

On the basis of media and media technologies

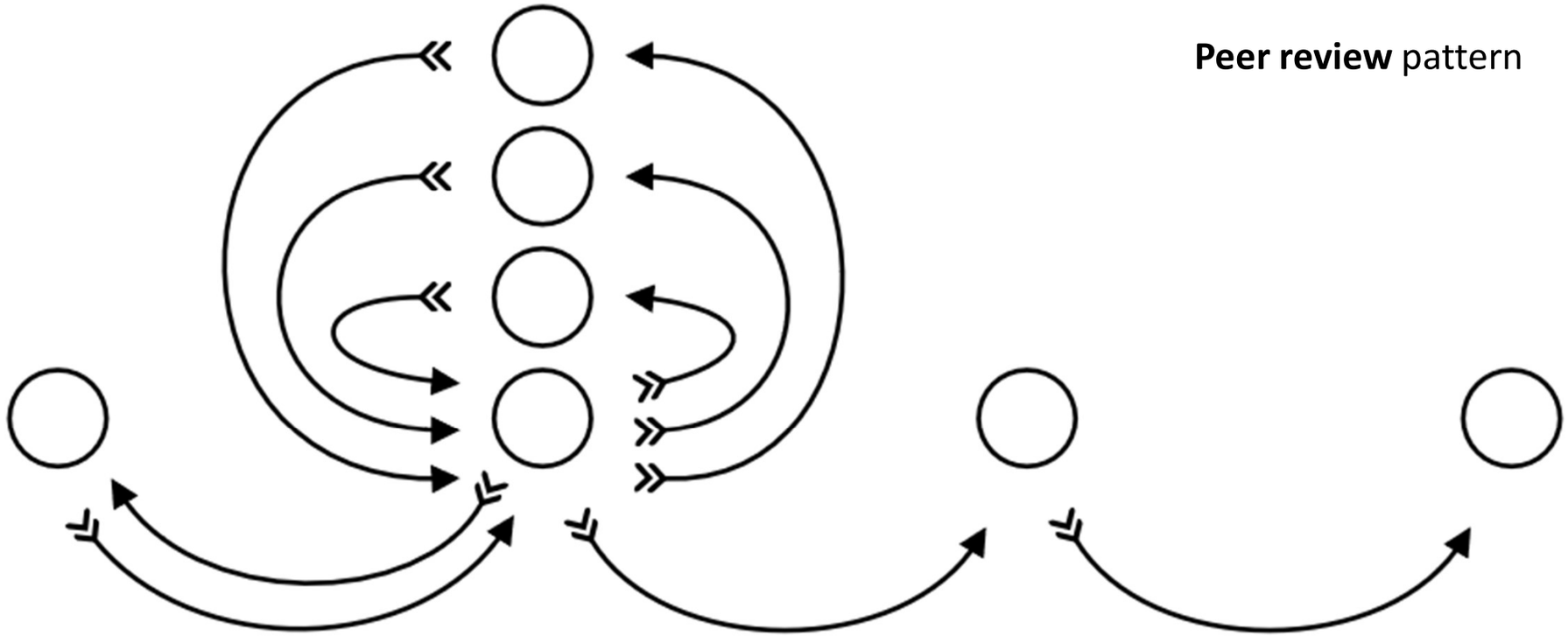


Public release and dissemination



- Information “for internal use”
- Information you are obliged to keep close
- Plan Z (the one you hope you never have to read)

“Real” workflows?



Peer review pattern

Whatever the factors and scale, workflow should be measurable:

E.g.: $\left(\frac{6\ articles}{issue}\right)\left(\frac{3\ peer\ reviews}{article}\right) = 18\ peer\ reviews/issue$

Declarative markup

What have we learned from its successes?

- Your system is someone else's subsystem
- Declarative markup adds (a particular kind of) value
- Data acquisition is hard
- Activities are supported by incentive structures
- Quality is defined within context
- Evolution happens by little revolutions

What do we see when we look at success?

Complex problems
Standards and community
Simple, ubiquitous tools

… where semantics are transparent across layers …

Content and format are distinguished …

… but they echo and reflect each other …

So … why *does* everyone still use office documents?

Answer: What's the domain expert's
best available tool
for on-the-fly data modeling?

Step 1: sketch, mockup or worked example

Step 2: form, spreadsheet or document template

Step 3: iterate under load

Templates and canonical examples encode semantics (or we wouldn't bother)

The information silo ("what silo?") doesn't feel confining till later

Systems description

The basis of any accounting for a system

is a description or representation of that system

- Declarative markup works
- But the reality (out there) is more complex:

  "Complete" and "correct" often means "looks good"

  So: office documents with and without templates

  Proliferation of *ad hoc* designs and approaches

  Plus a vast amount of machine-readable data already available in great mix of formats

While the system description may be knowable, the system itself is not …

… we need a *language*

(Micro) publishing is nothing but …
… content engineering …

“Traditional” publishing

Writers

Editors

Peer reviewers

Copy editors

Fact checkers

Bibliographers

Catalogers

Managing editors

Marketers

Indexers

Designers

Data conversion specialists

“Paperwork” oriented RMF

Systems architects and developers

Systems analysts and integrators

Policy analysts

Operational staff

Trainers

Reviewers

Assessors

Project Managers

Owners and proprietors

- Systems are hybrid systems of systems
- “Paper” is now electronic but the “paper trail” was never even paper …

It’s 2020 and hypertext is real

Only problem is … PDF and office documents?
This is a limiting factor for scalability

Fortunately: *we know an approach that works*

- Community-centric standards development
- Openly specified non-proprietary data formats
- Providing a common basis for data interchange among independent organizations
  - “Independent” means “not always all the same rules”
  - Multi-party not just two-party exchanges
- Commodity tools and toolkits
- Support from commercial solution providers and markets
- Addressing a wide range of users in different roles with different kinds of expertise

Lightweight data modeling in two layers

Initiatives at NIST ITL

Metaschema

- Small schema language + extras
- XML, JSON, YAML, no problem
- Tools and tooling
- Documentation production
- Constraint checking
  - including nested and contingent constraints and constraint sets
- Usable by content experts

OSCAL

The Open Security Controls Assessment Language

- Set of document models
- Designed by practitioners
- Describing operational, interlinked document sets
- Defined by a library of metaschemas

Testing the models in the field

FedRAMP: The Federal Risk and Authorization Management Program

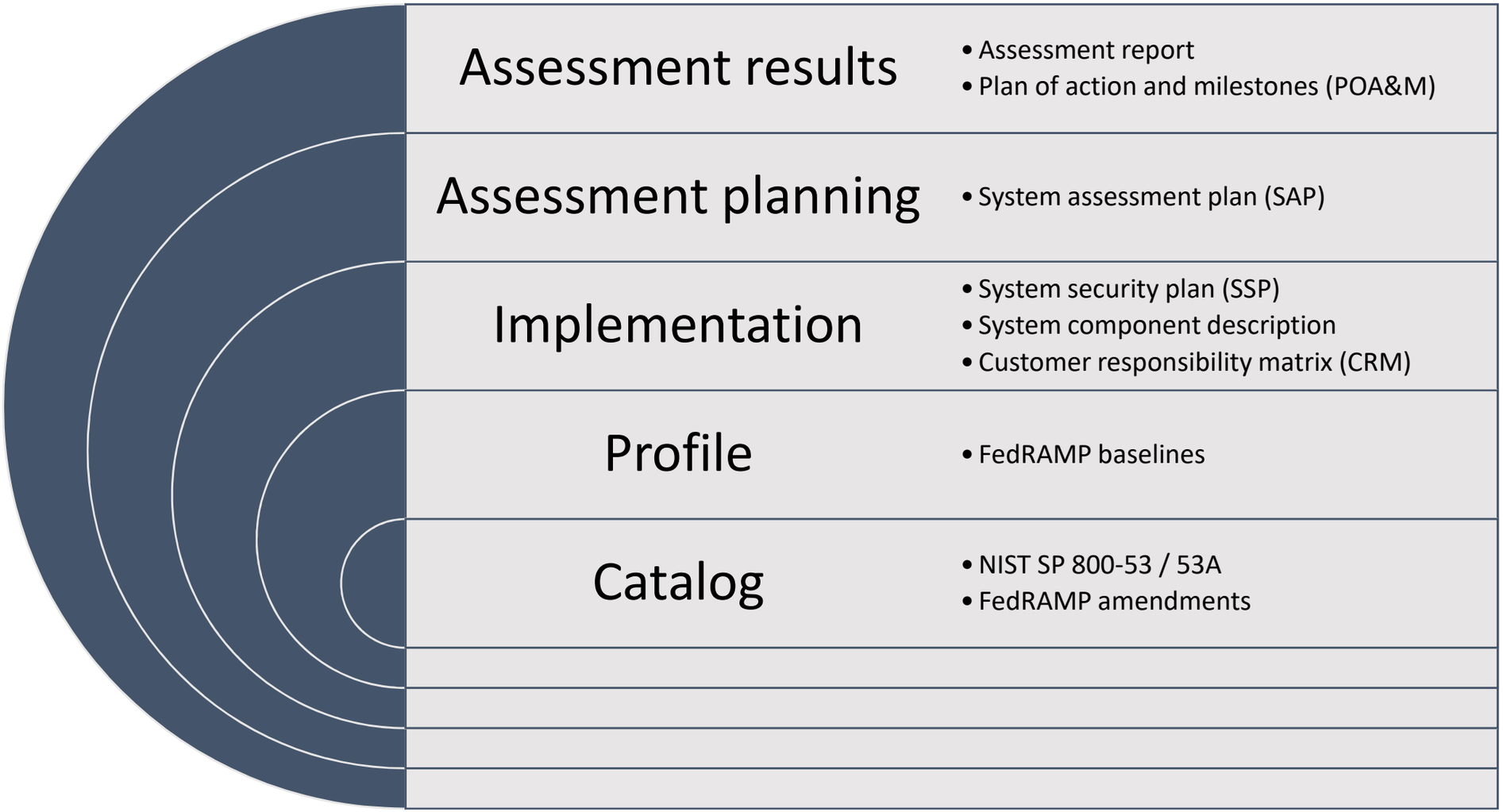
A US federal government program under the General Services Administration, Office of Management and Budget (GSA OMB)

"Provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services" https://fedramp.gov/about/

- Helping government agencies and programs to meet security and RMF*-related requirements for systems acquisition and operation
- *FedRAMP is using OSCAL to define models to describe the documentary components of their authorization and assessment workflows*

* RMF is the Risk Management Framework https://doi.org/10.6028/NIST.SP.800-37r2

OSCAL Architecture (FedRAMP application space)



Reinventing wheels?

Some things never change

- Today we see a mix – tomorrow?
  - office documents (word processors, spreadsheets)
  - graphic artifacts and attachments
  - *sometimes* structured data sets
- Information shelf life – data longevity – ranges widely
  - Some of the material is canonical and stable, with version control
  - Other material is very ephemeral
    - Goes out of date, and/or subject to change without notice
    - May be a temp / throwaway / mockup / early draft of a “real” document

Significant players already have their solutions

The present gap is not how to achieve security (for those who can afford it)

But how to support everyone’s achieving it more easily with more help from one another

There are great opportunities here for nimble participants

Able to work across boundaries

What are the lessons?

Your system is someone else's subsystem

- It might not be necessary to solve the entire problem at once

Declarative markup adds (a particular kind of) value

- Whether it is XML or JSON may not ultimately matter
- We need both markup languages and object notations
- With seamless interchange across and between

Data acquisition is hard

- Data acquisition will always be hard
- (Until all information is captured at the point of creation?)

Activities are supported by incentive structures

- Early benefits can be shared
- So can later benefits
- Meanwhile, what incentives can we offer for *learning*?

Quality is defined within context

- Constraints are reflected in opportunities

Evolution happens by little revolutions

- *It might not be necessary to solve the entire problem at once*

The capability is not the stack


Office documents
Spreadsheets, tables, grids, lists, digests
Structured data

Thank you!

Questions, observations, advice?

Acknowledgements:
Michaela Iorga
Josh Lubell
Brian Ruf
Dave Waltermire

National Institute of Standards and Technology
U.S. Department of Commerce