SIEMENS
Ingenuity for life

Version 1.0
April 2019

Siemens' experiences with 'Scapolite', a YAML+Markdown-based alternative to XCCDF

(presented @NIST SCAP v2 Workshop April 30th to May 2nd 2019)

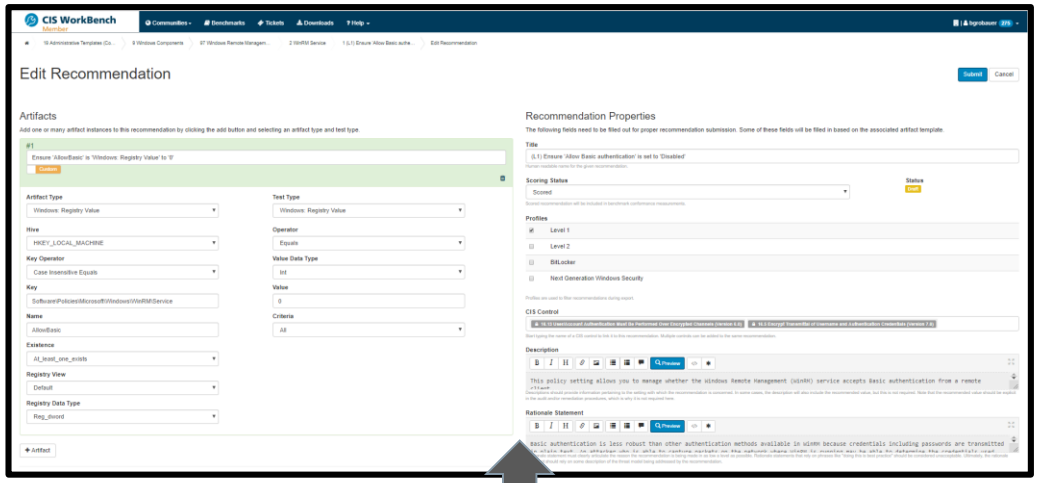
Dr. Bernd Grobauer, Siemens CT RDA ITS

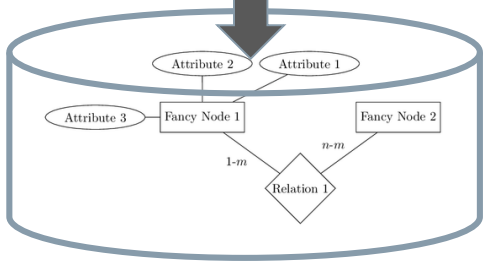
© Siemens AG 2019

Two (and *only* two) publishers of SCAP content that support community-driven authoring and maintenance

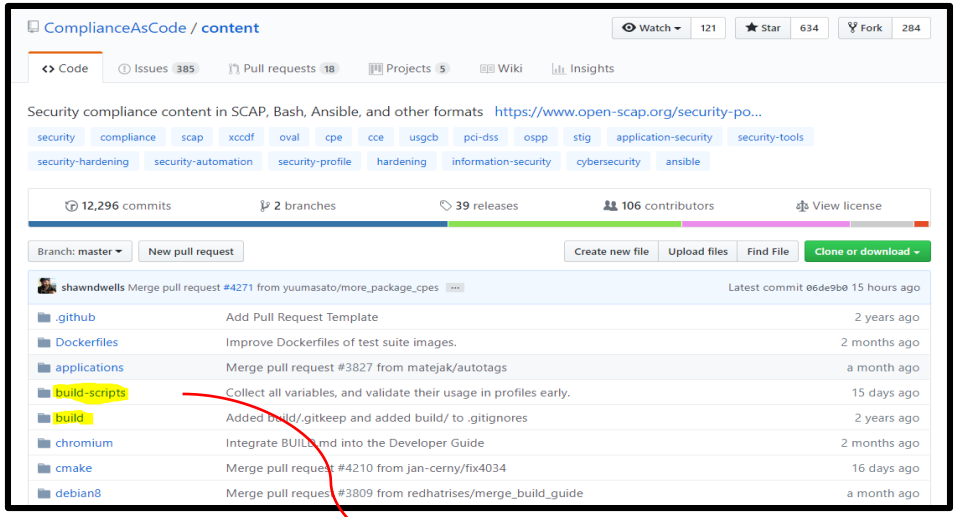
SIEMENS
Ingenuity for life

CIS





OpenSCAP



> `make all`

Both approaches have in common:

- Found it impossible to author and maintain SCAP as „SCAP proper“
- Chose to work with an internal“ representation of data is **not** SCAP:
  - CIS: database schema with semantics implicit in code of workbench application
  - OpenScap: mixture of file formats and file-system layout, with semantics implicit in code of build process

© Siemens AG 2019

Siemens‘ approach: Inspired by OpenSCAP (THANKS!!!), but separating content and build process via explicit semantics

SIEMENS
Ingenuity for life

Siemens CT RTC ITS

B. Grobauer
Siemens AG
June 23, 2017

Scapolite: YAML- and Markdown-based Authoring and Manipulation of IS Rules and Benchmarks

draft-grobauer-scapolite-latest

Abstract

The SCAP standard for communicating machine-readable security benchmarks [SCAP_1_2] has been around since 2009. There has been significant but by no means overwhelming take-up of SCAP. What is most noticable, is the huge disparity between authors and consumers of SCAP. Authoring of SCAP content is done mostly by a few organizations, mainly governmental as well as the occasional vendor and non-for-profit organization. But even though there are many more organizations that author and maintain IT security rules, almost no organization uses SCAP for this purpose. This is, because authoring SCAP content is extremely complicated and cumbersome. Tools are hard to come by and have mostly proven inadequate for all but the most basic usage. SCAP's XML also does not lend itself for direct authoring on text-file basis, no matter whether a special XML editor is used, or not.

The most promissing approach towards writing SCAP standard has been developed by the OpenSCAP project: SCAP content is devided into pieces of simplified XML stored in single files. Colaborative editing is done using git as version control system – SCAP content is then produced by a combination of scripts and tools that collect the XML pieces and then transform and combine them into a SCAP data stream.

Scapolite takes inspiration from OpenSCAPs approach, but goes one step further: rule collections and rules are specified in a combination of YAML and Markdown, thus putting content into a form that can easily be edited directly with a text editor, but at the same time is machine readable and thus can be read by tools for transformation into SCAP or other relevant formats.

(c) 2017 Siemens CT RDA ITS

1. Introduction

Table of Contents

Format Definition
providing Semantics

SCAPOLITE

Implementation of Tooling

`scaptain` command line tool

`SFeRA` web application for
employee access to
Siemens IS Policy Framework

SFeRA

© Siemens AG 2019

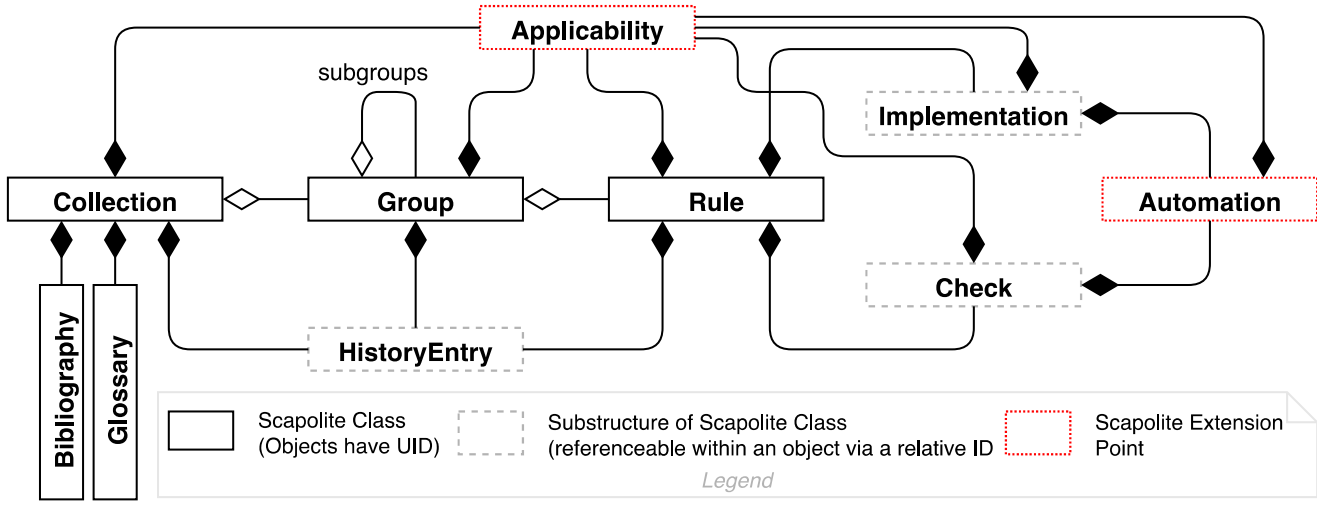
Scapolite Rule Example
Introductory Example taken from Scapolite Documentation

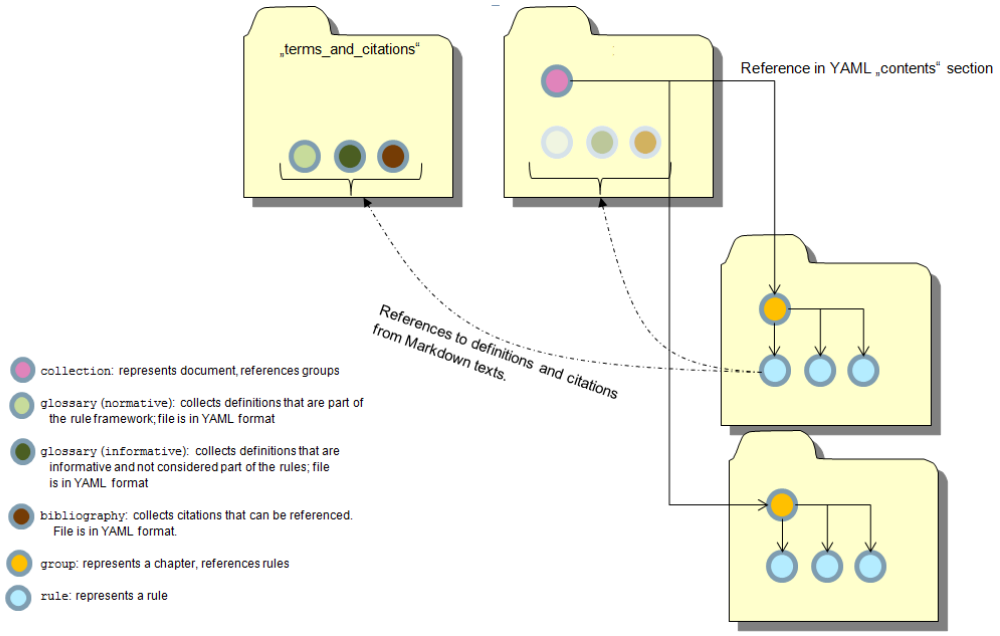
SIEMENS
Ingenuity for life


```
scapolite:
    class: rule
    version: '1.0'
id: C085611074
id_namespace: com.siemens.cert.scapolite.example_benchmark
title: An example rule
rule: Do as I say, not as I do.
rationale: |
    There are always example of policy/rule makers who do not conform to their own
    rules. Nevertheless, many of their rules are sensible and **MUST** be obeyed.
implementations:
  - relative_id: '01'
    title: Just do it youself
    description: |
        Carry out the following steps:

          - Do this
          - Do that
  - relative_id: '02'
    title: Get people to do it
    description: |
        Carry out the following steps:

          - Check whether people are doing it
          - If not: **shout** at them
          - Repeat
```

Substructures with machine-readable automations of an implementation can be added to an implementation or a check.

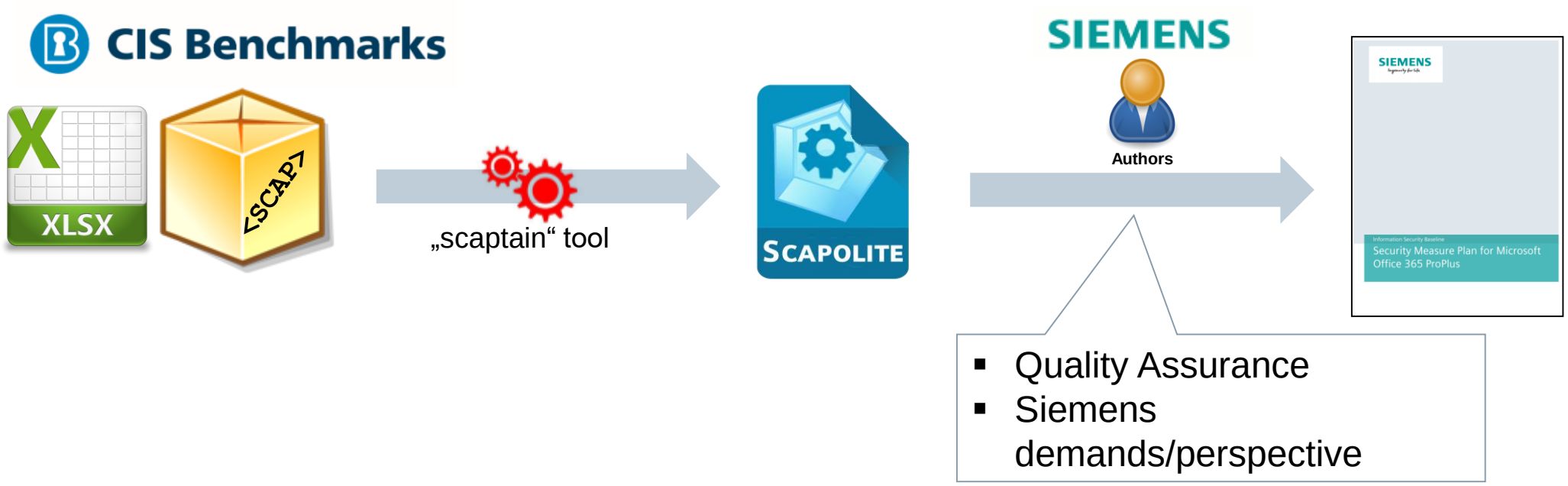
Class Diagram and exemplary file layout





© Siemens AG 2019

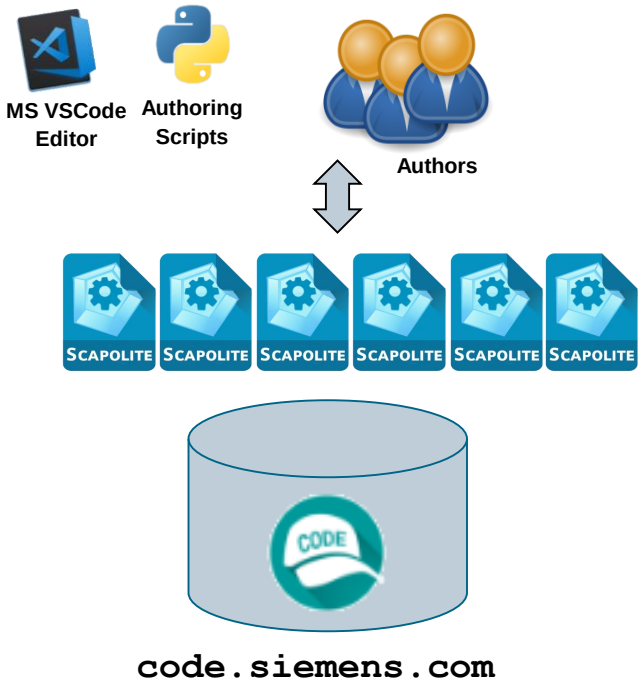
Leveraging externally available benchmarks with Scapolite

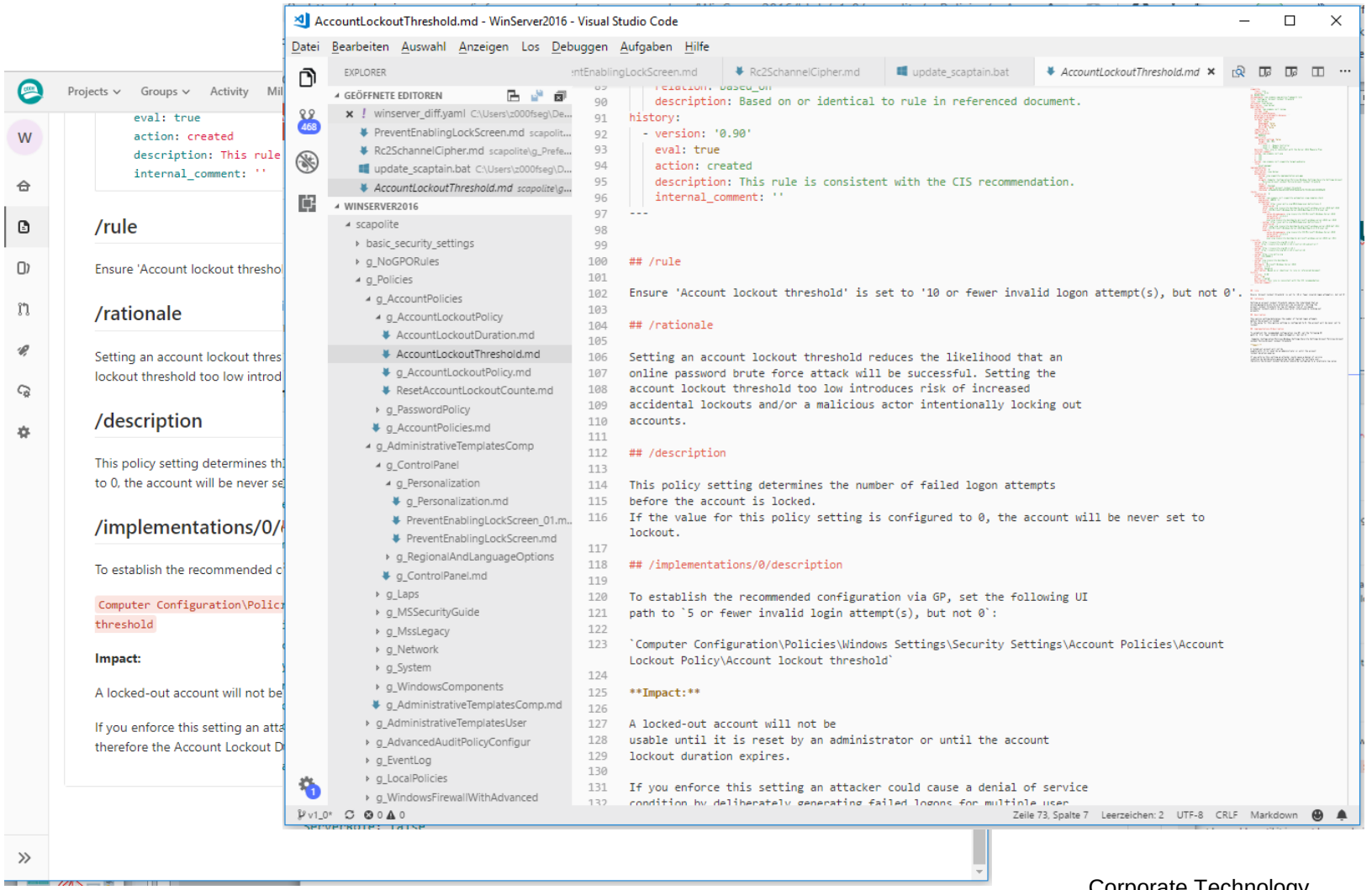


© Siemens AG 2019

Scapolite: Taking a leaf out of the book of managing code

SIEMENS
Ingenuity for life

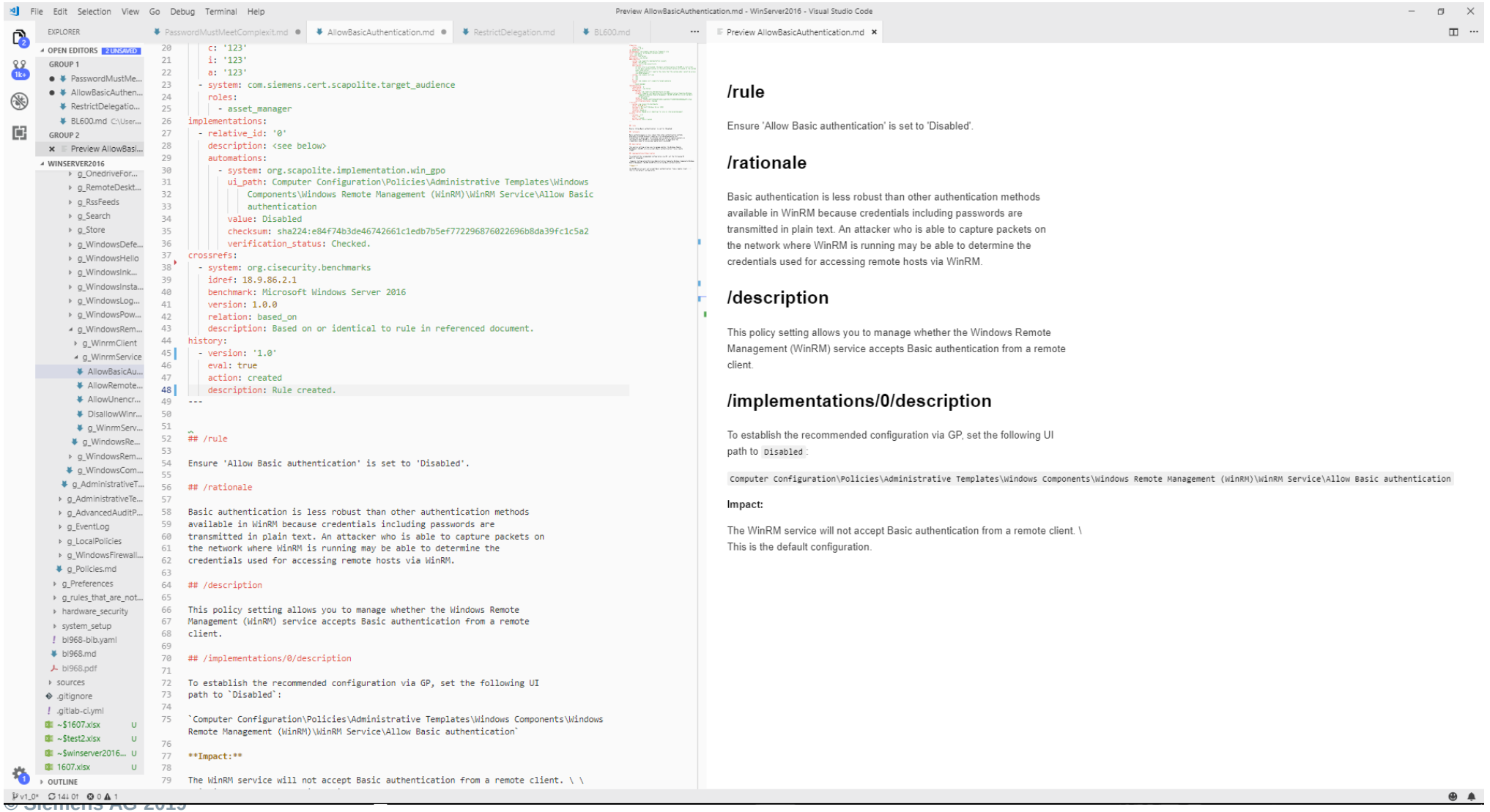




© Siemens AG 2019

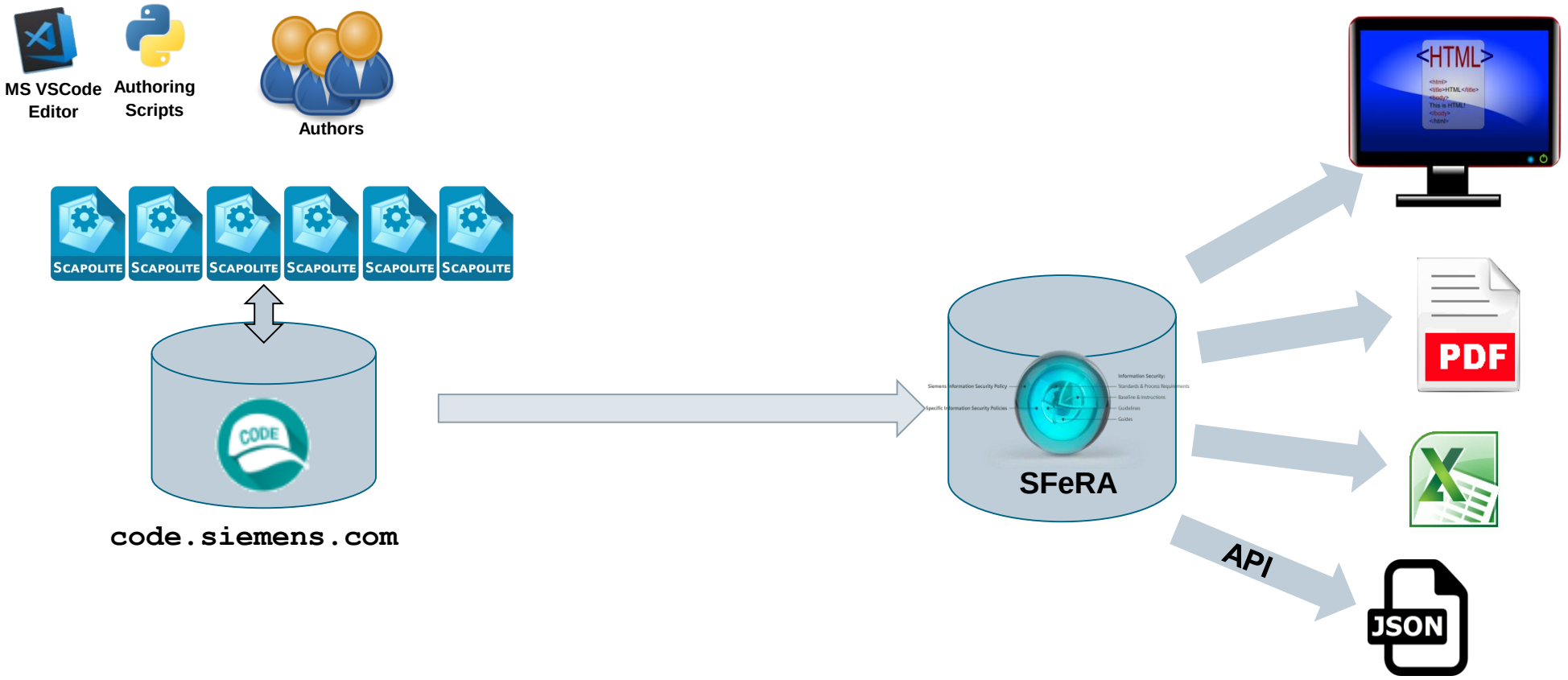
Example: Editing a Scapolite document

Factoring out Markdown content from YAML fields allows use of editors‘/gitlab‘s preview feature and avoids problems with „double indentation“ of YAML & Markdown



© Siemens AG 2019

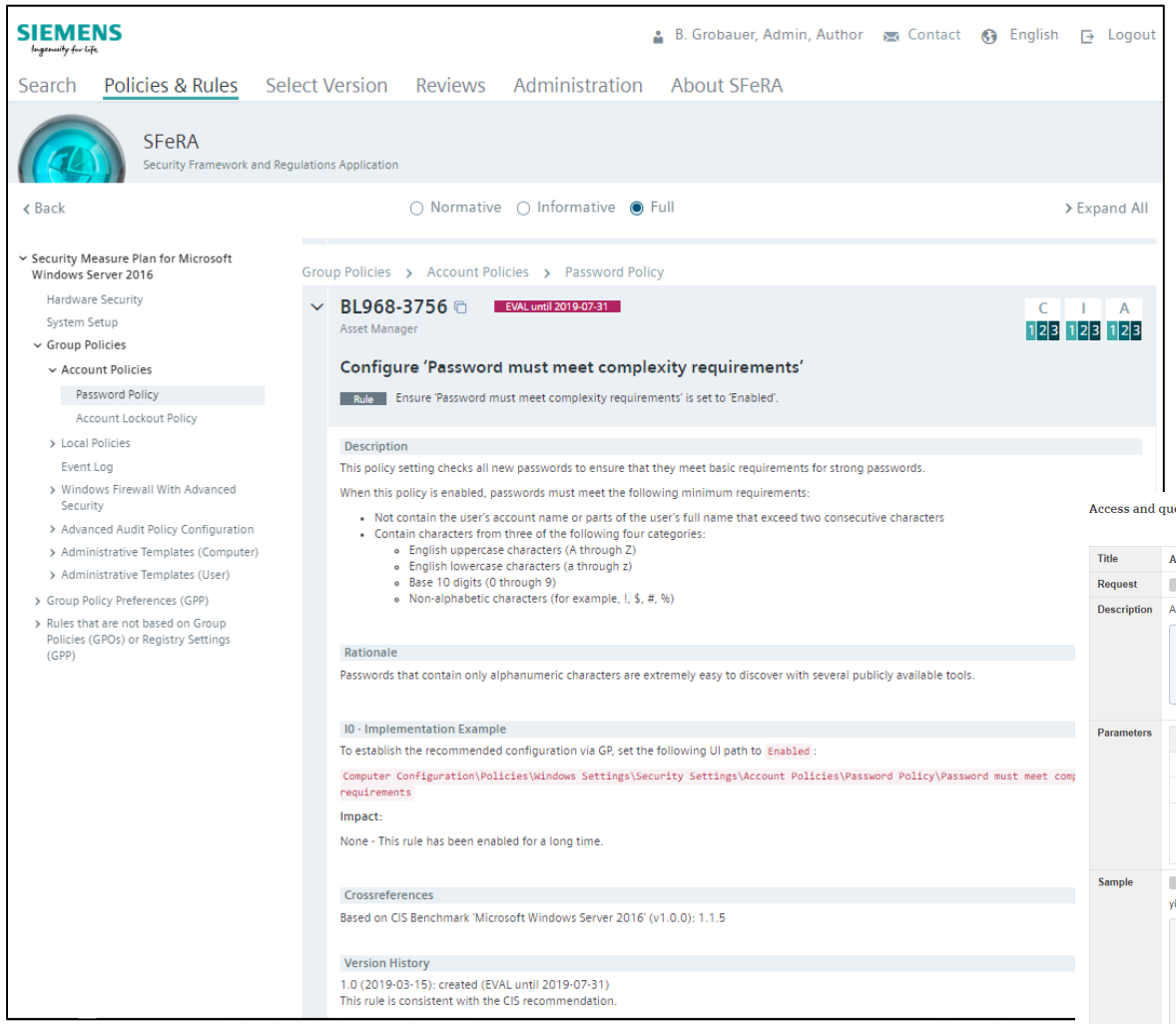
SFeRA: From plain text to many formats

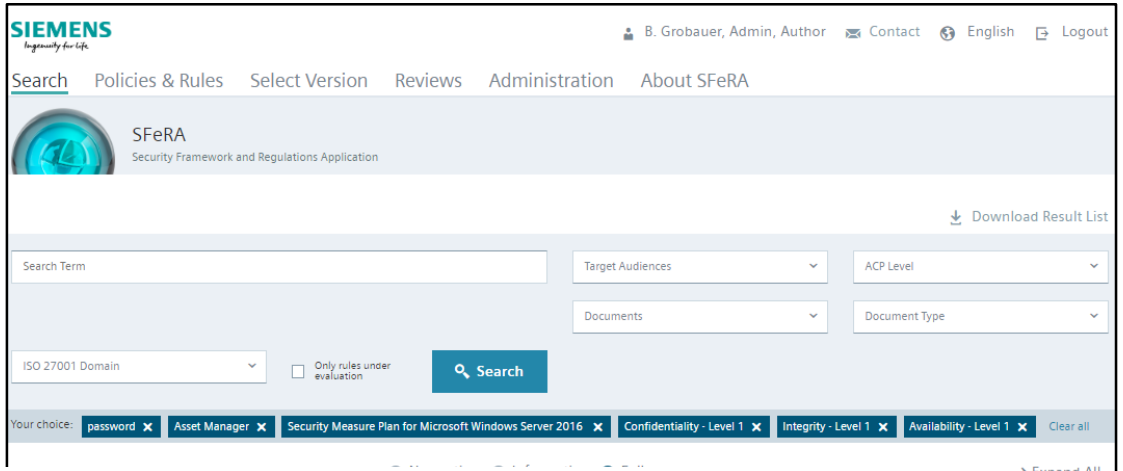


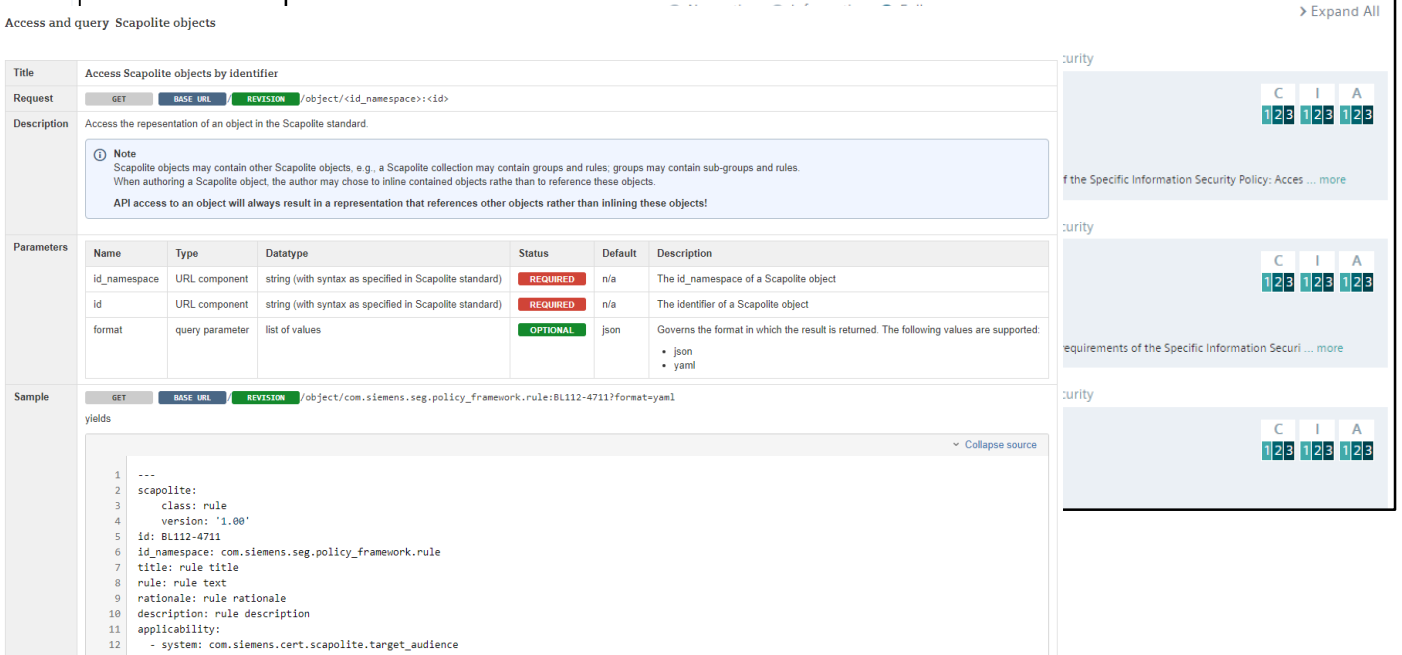
© Siemens AG 2019

SFeRA

SIEMENS
Ingenuity for life



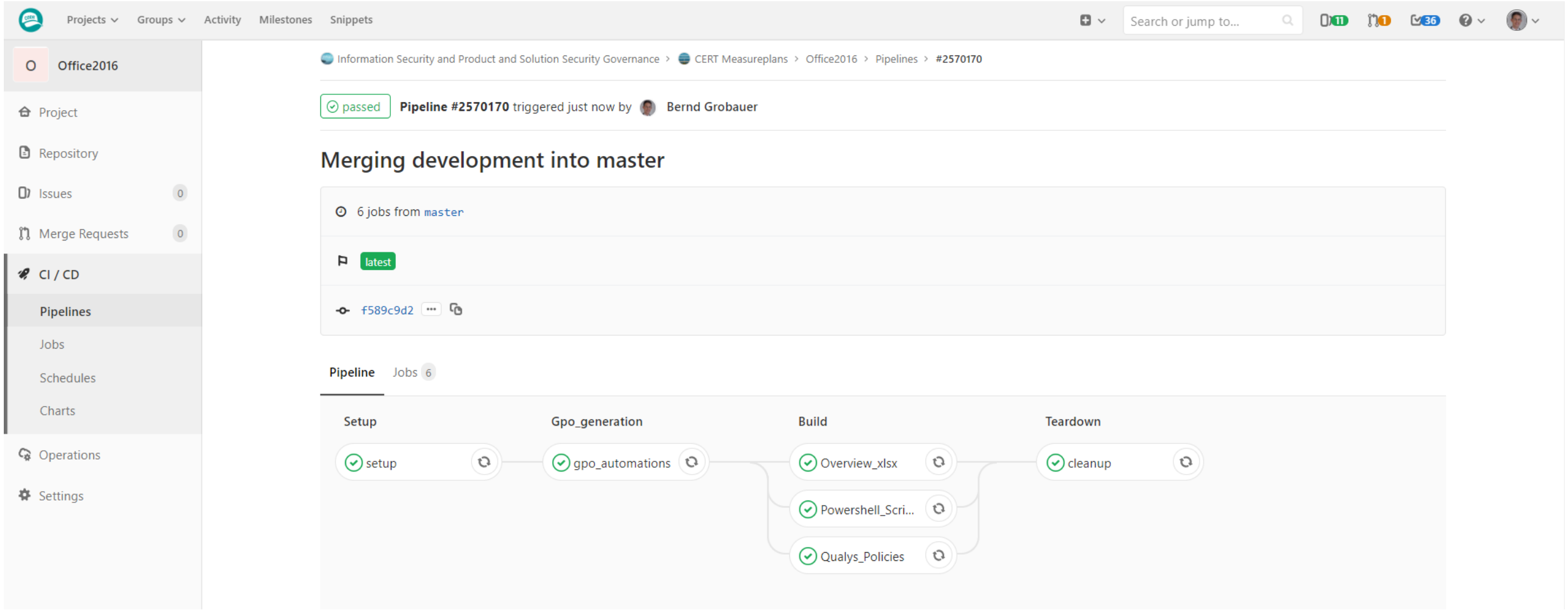




© Siemens AG 2019

Using Gitlab‘s CI features for DevOps-inspired approach towards maintaining security baselines

SIEMENS
Ingenuity for life



© Siemens AG 2019

Conclusion

SIEMENS
Ingenuity for life

- There is huge demand for machine-readable security baselines, yet it seems that most organizations merely consume SCAP content by one of the „big three“ (IASE, CIS, OpenSCAP) rather than producing their own SCAP content
- One probable reason: authoring and maintaining content in „SCAP proper“ is almost impossible
- Proposed solution: SCAP v2 must define standard formats that truly allow a „security-as-code“ approach
- Internal usage of Scapolite for all new IS Policies published within Siemens in the past 1.5 years shows that Scapolite is a format that supports the „security-as-code“ approach

  (If there is interest, it might be possible for us to publish Scapolite (e.g., as IETF RfC) and (2) provide central parts of scaptain as open-source code.)
- SCAP v2 should also try to put more focus on automating also the implementation rather only the check (more about this topic in tomorrow‘s talk by Patrick Stöckle)

© Siemens AG 2019