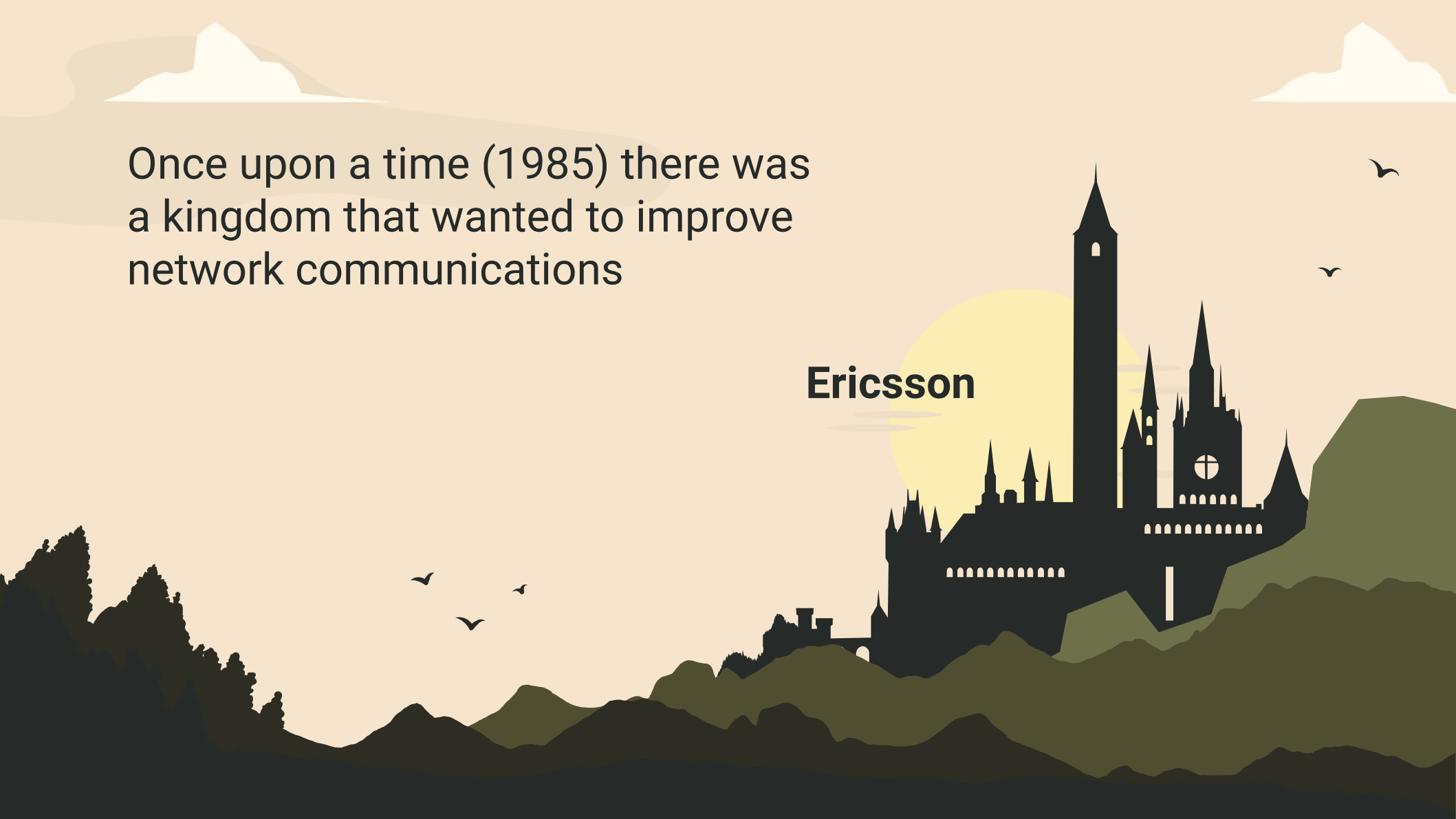
First Steps With ORT
An EEF Experience

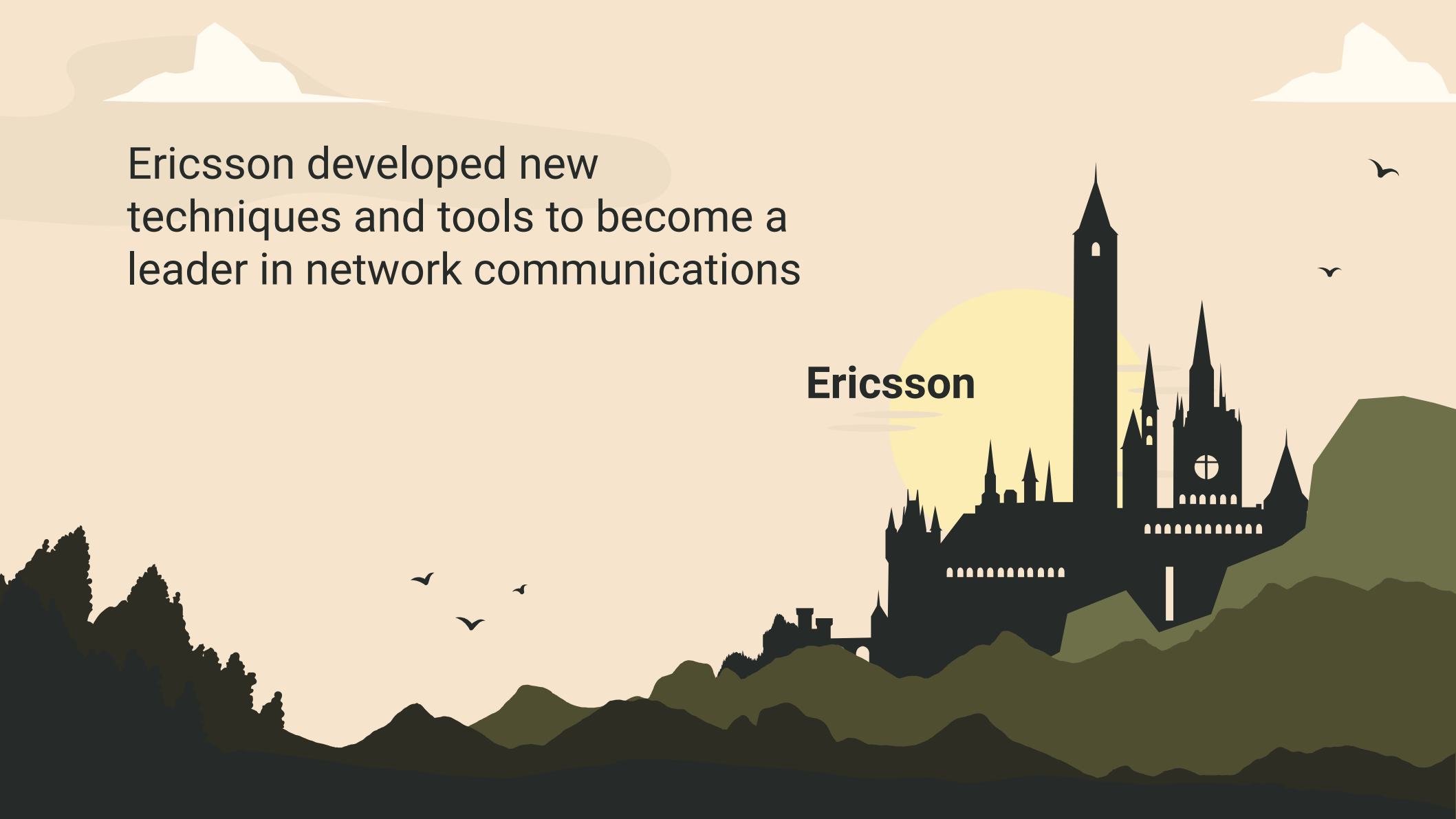
Kiko Fernandez-Reyes

Once upon a time (1985) there was a kingdom that wanted to improve network communications

Once upon a time (1985) there was a kingdom that wanted to improve network communications



Ericsson developed new techniques and tools to become a leader in network communications



1985 - 1998

1985 - 1998

Erlang/OTP

Erlang/OTP

01

Concurrent

02

Distributed

03

Fault Tolerant

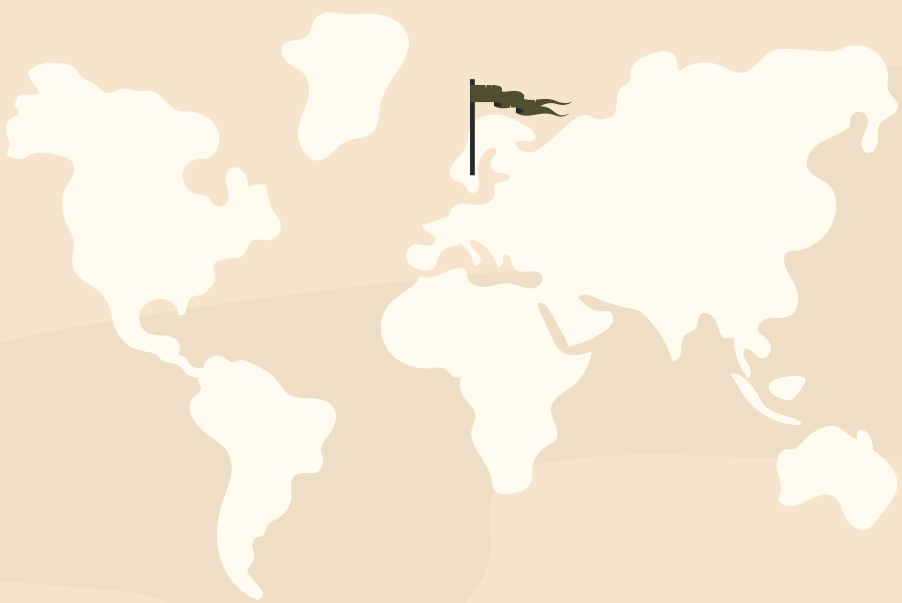
1998

1998

Ericsson decided to share their technology with the world

In search of lost lands

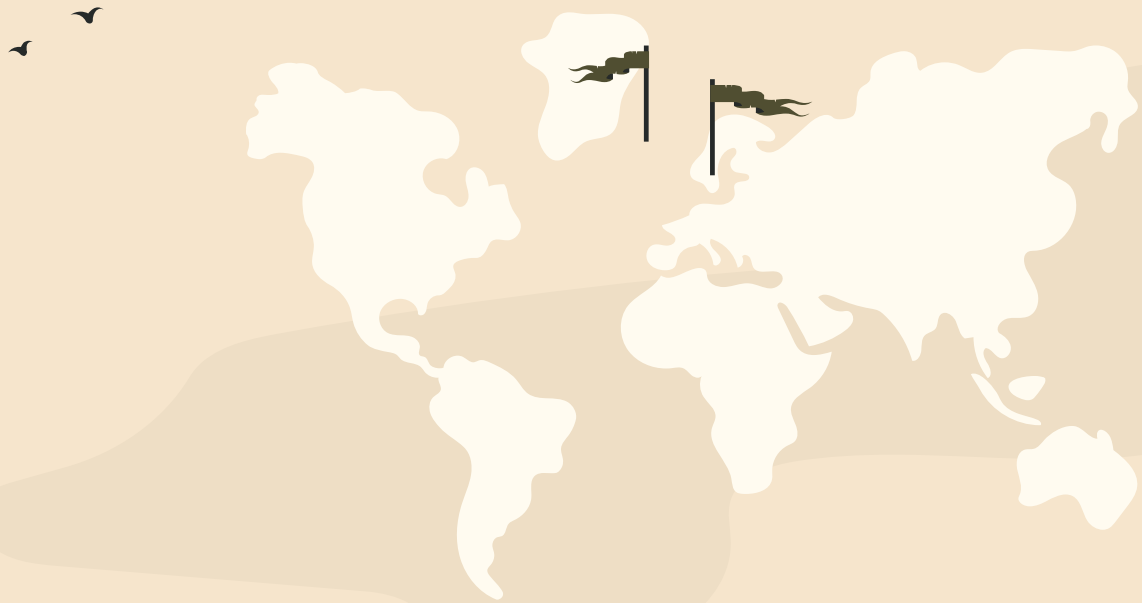
In search of lost lands



Kivra

Klarna

In search of lost lands

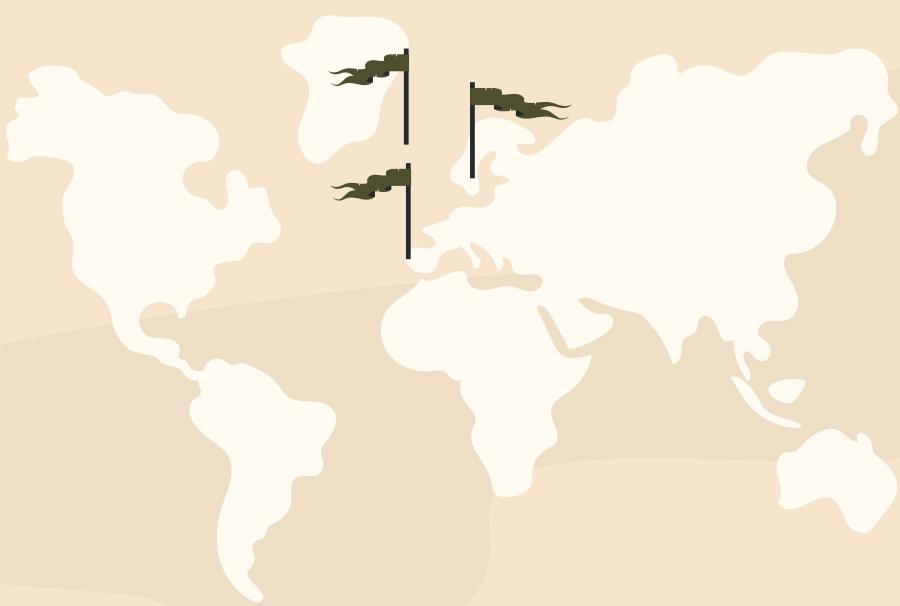


WhatsApp

Kivra

Klarna

In search of lost lands



WhatsApp

Kivra

Klarna

Academia

In search of lost lands

WhatsApp

Kivra

Klarna

Meta

Academia

In search of lost lands

Cisco

Meta

Academia

WhatsApp

Kivra

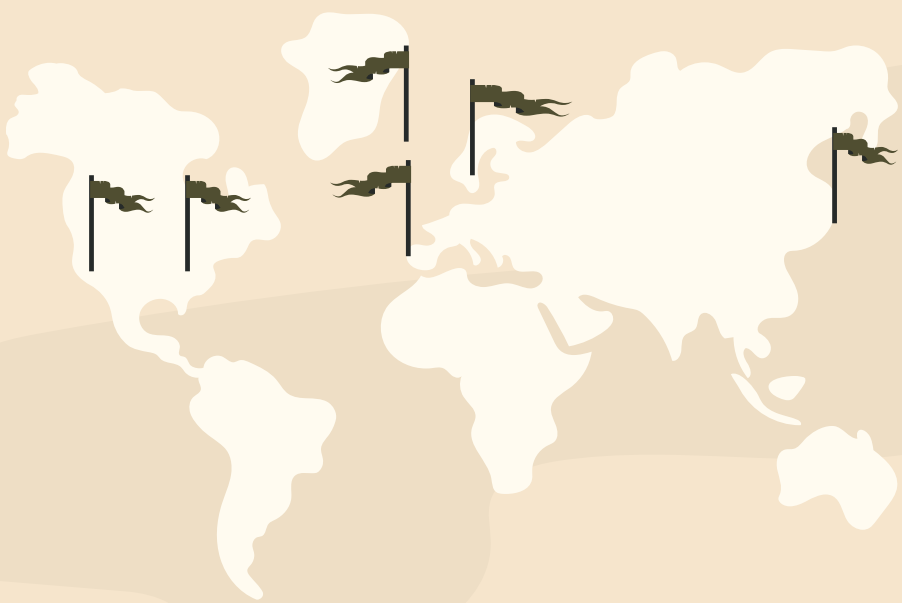
Klarna

In search of lost lands

Cisco

Meta

Academia



WhatsApp

Kivra

Klarna

Rakuten

In search of lost lands

Cisco

Meta

Academia

WhatsApp

Kivra

Klarna

Rakuten

In search of lost lands

Cisco

Meta

Academia

WhatsApp

Kivra

Klarna

Rakuten

In search of lost lands

Cisco

Meta

Academia

WhatsApp

Kivra

Klarna

Rakuten

In search of lost lands

Cisco

Meta

Academia

WhatsApp

Kivra

Klarna

Rakuten

In search of lost lands

Cisco

Meta

Academia

WhatsApp

Kivra

Klarna

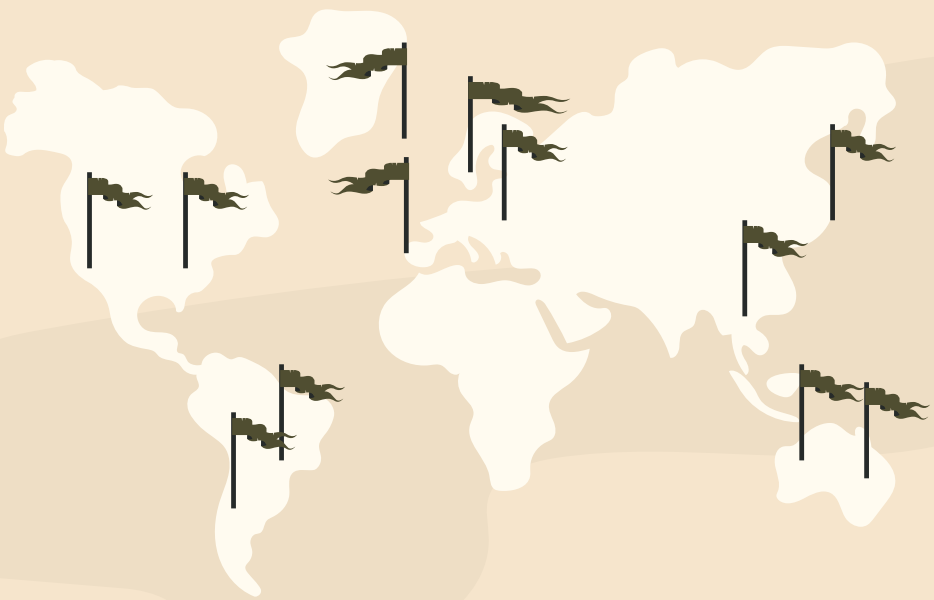
Rakuten

In search of lost lands

Cisco

Meta

Academia



WhatsApp

Kivra

Klarna

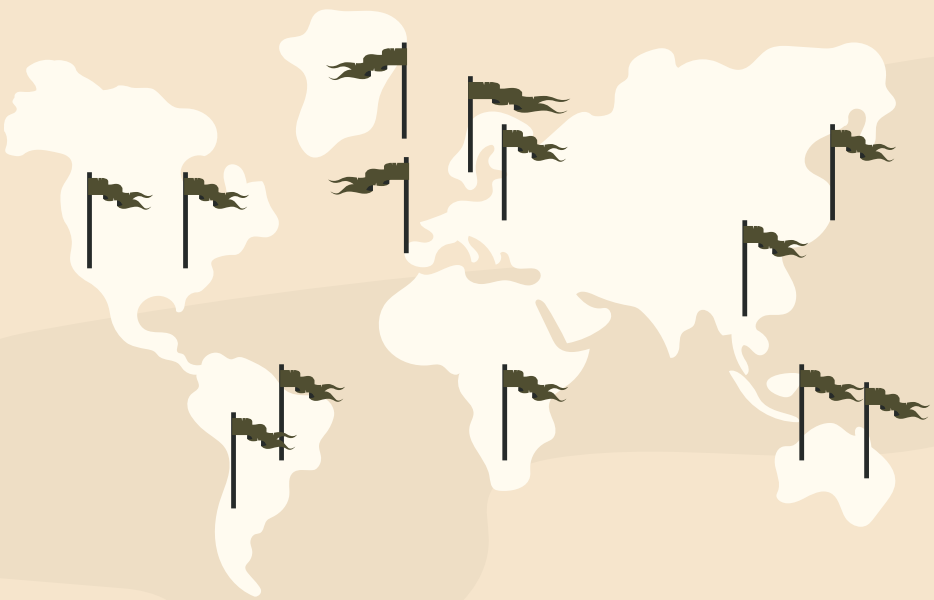
Rakuten

In search of lost lands

Cisco

Meta

Academia



WhatsApp

Kivra

Klarna

Rakuten


Ericsson
Erlang/OTP is not just Ericsson


WhatsApp


Academia
WhatsApp


Cisco
Academia
WhatsApp


Cisco
Academia
WhatsApp
Elixir


Cisco
Academia
Gleam
WhatsApp
Elixir


Cisco
Academia
Gleam
WhatsApp
Elixir
Erlang
Ecosystem
Foundation

Erlang Ecosystem Foundation

Cyber Resilience Act

Cyber Resilience Act

Cybersecurity

Cyber Resilience Act

Cybersecurity

Rules to ensure cybersecurity of products with digital elements

Cyber Resilience Act

Cybersecurity

Rules to ensure cybersecurity of products with digital elements

Products shall be made available with a **secure by default** configuration

Cyber Resilience Act

Cybersecurity

Rules to ensure cybersecurity of products with digital elements

Products shall be made available with a **secure by default** configuration

Identify and document **vulnerabilities**, including a **software bill of materials** in a commonly used and machine-readable format

Cyber Resilience Act

Cybersecurity

Rules to ensure cybersecurity of products with digital elements

Products shall be made available with a **secure by default** configuration

Identify and document **vulnerabilities**, including a **software bill of materials** in a commonly used and machine-readable format

Apply **effective** and regular **tests** and **reviews** of the **security**

Context in relation to ORT

Context in relation to ORT

Licenses and Copyrights

Context in relation to ORT

Licenses and Copyrights

Contributors

Context in relation to ORT

Licenses and Copyrights

Contributors

Vendor

Context in relation to ORT

Licenses and Copyrights

Contributors

Vendor

Compliance

Context in relation to ORT

Licenses and Copyrights

Compliance

Contributors

Vulnerability

Vendor

Context in relation to ORT

Licenses and Copyrights

Compliance

Contributors

Vulnerability

Vendor

Files:

Erlang	11,486
Elixir	1,214
Gleam	3,248

First Steps using ORT in Erlang/OTP

11,000+ Files

Vendor

11,000+ Files

Vendor

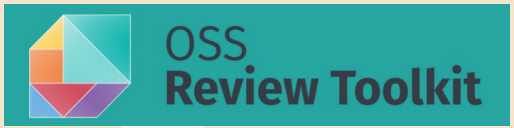
```
[AsmJit]

* Info
  * SPDX-License-Identifier: Zlib
  * Library: AsmJit
  * Git Repository: https://github.com/asmjit/asmjit
  * Commit: 029075b84bf0161a761beb63e6eda519a29020db
  * OTP Location: erts/emulator/asmjit

Copyright (c) 2008-2025 The AsmJit Authors

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

-------------------------------------------------------------------------------
[PCRE2]

* Info:
  * SPDX-License-Identifier: BSD-3-Clause
  * Library: PCRE2
  * Version: 10.45
  * Website: https://www.pcre.org
```

11,000+ Files

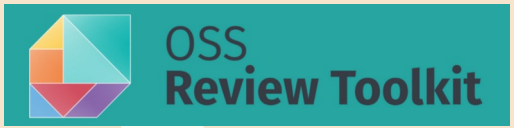
Vendor

Compliance

```
[AsmJit]

* Info
  * SPDX-License-Identifier: Zlib
  * Library: AsmJit
  * Git Repository: https://github.com/asmjit/asmjit
  * Commit: 029075b84bf0161a761beb63e6eda519a29020db
  * OTP Location: erts/emulator/asmjit

Copyright (c) 2008-2025 The AsmJit Authors

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

-------------------------------------------------------------------------------
[PCRE2]

* Info:
  * SPDX-License-Identifier: BSD-3-Clause
  * Library: PCRE2
  * Version: 10.45
  * Website: https://www.pcre.org
```



Compliance & Source SBOM



Compliance & Source SBOM

Threat

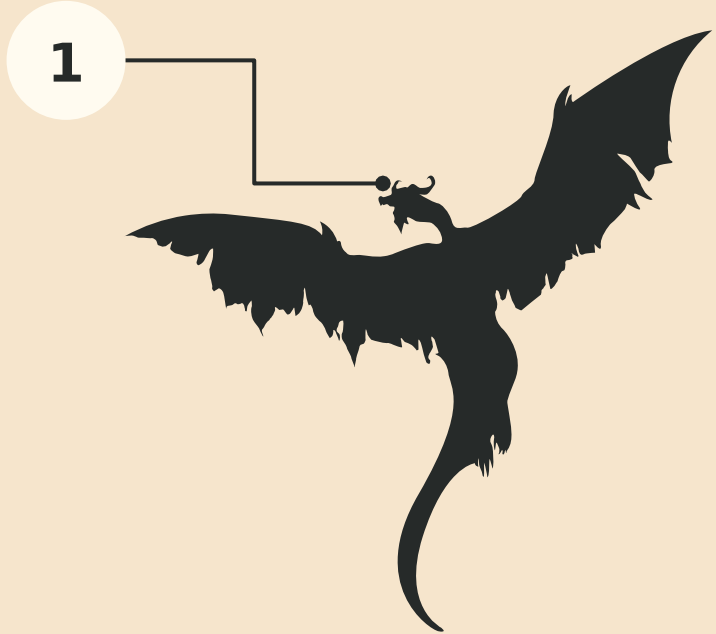
OSS Review Toolkit

Compliance & Source SBOM

Threat

OSS Review Toolkit

Compliance & Source SBOM



Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

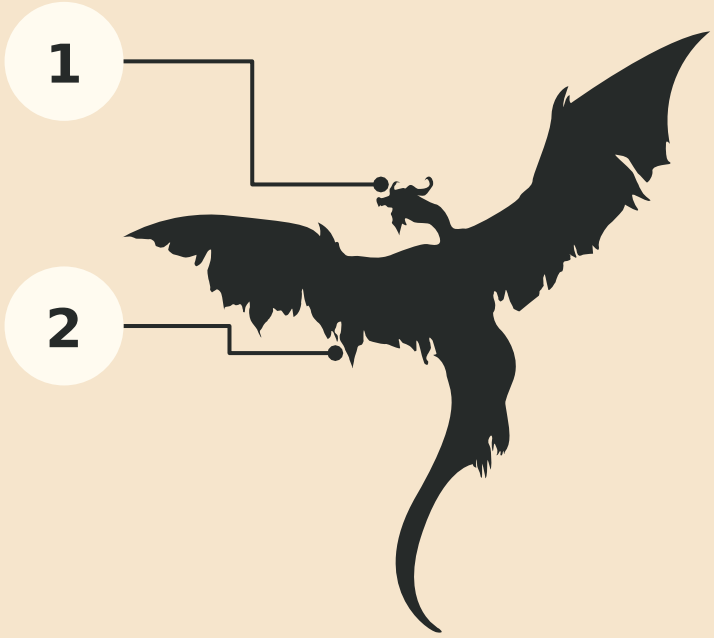
Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects



Threat

OSS Review Toolkit

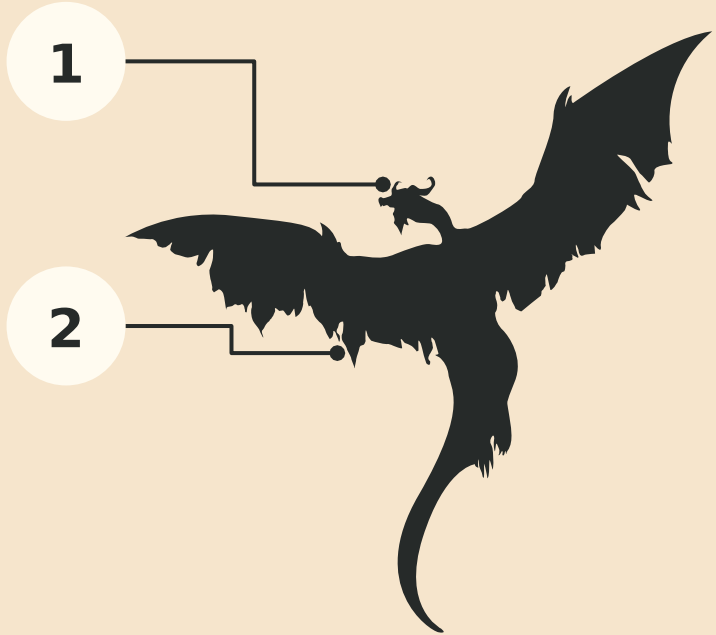
Compliance & Source SBOM

Analyser

Gets dependencies of projects

Downloader

Downloads dependencies to scan



Threat

OSS Review Toolkit

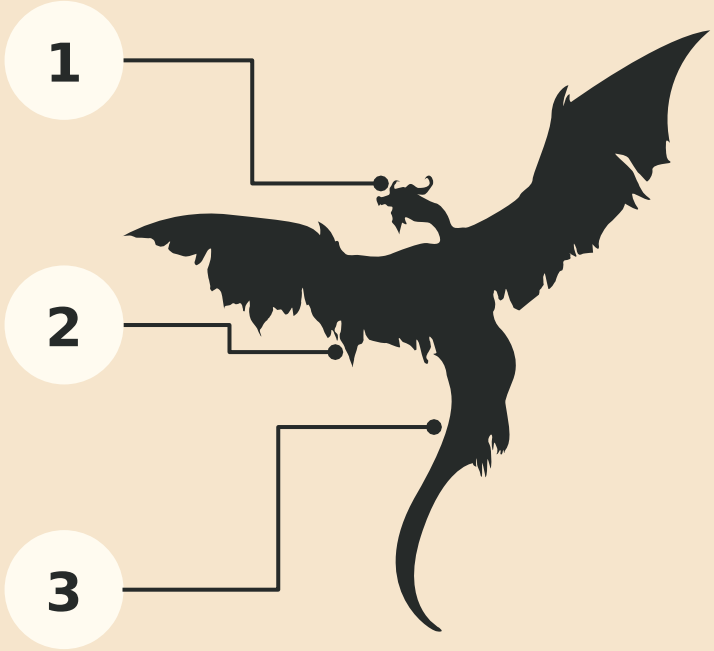
Compliance & Source SBOM

Analyser

Gets dependencies of projects

Downloader

Downloads dependencies to scan



Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

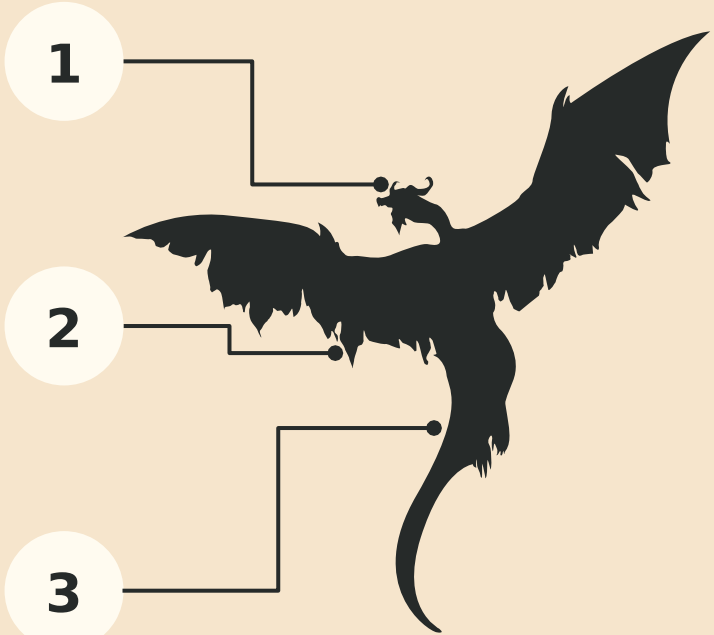
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

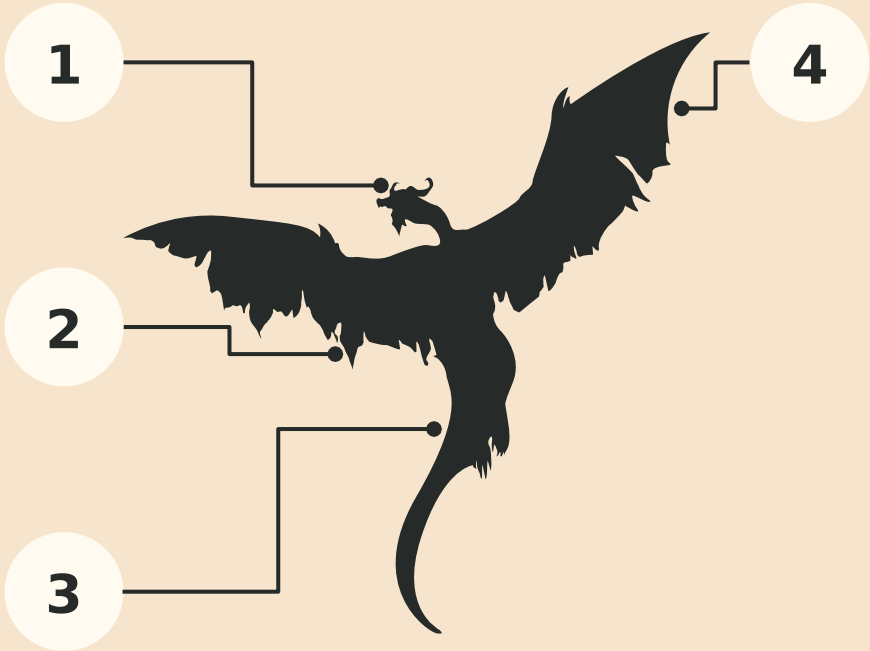
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

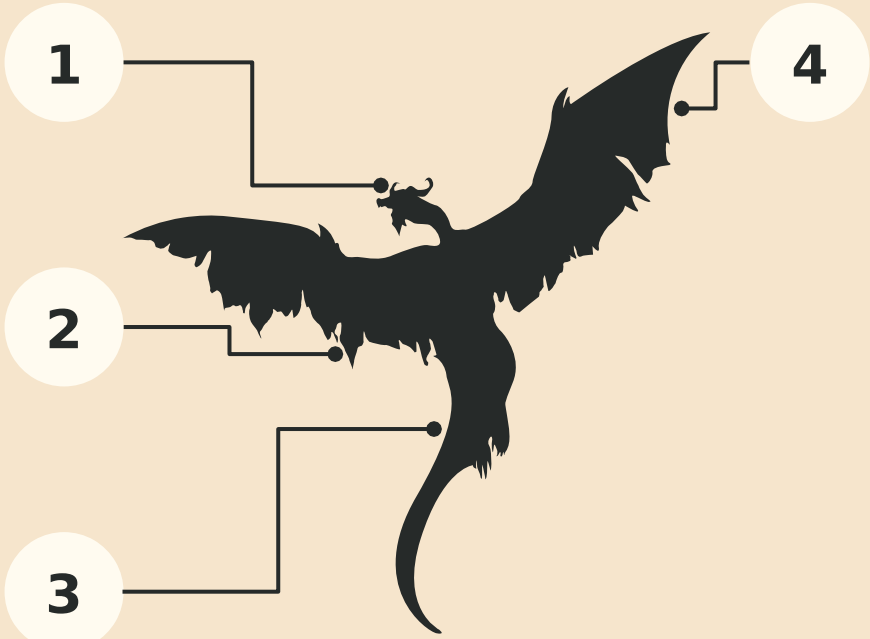
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

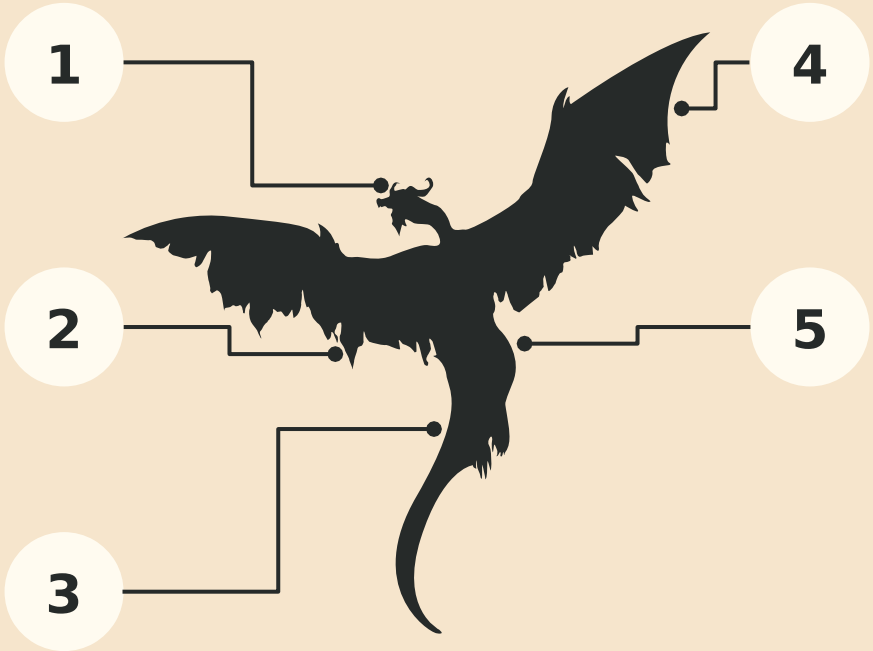
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

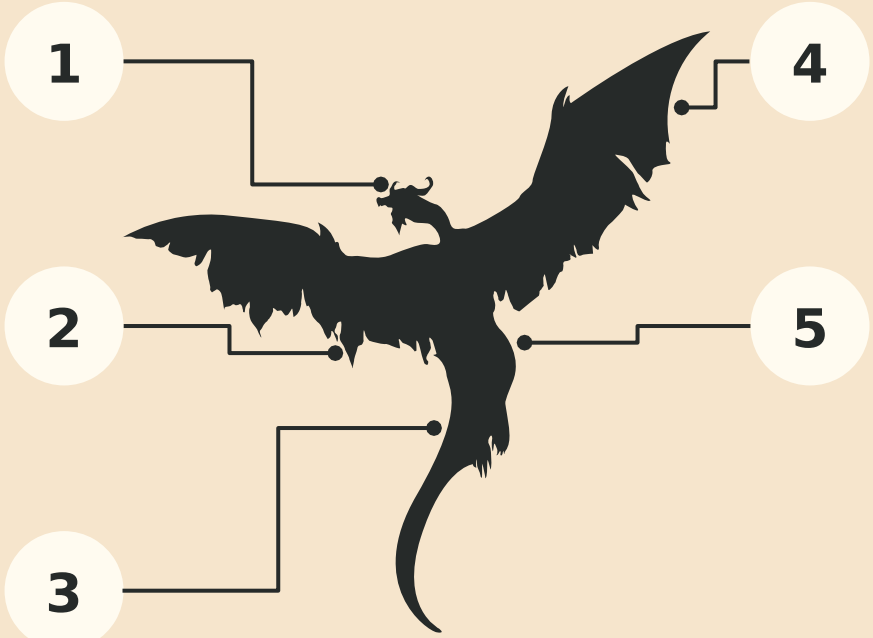
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

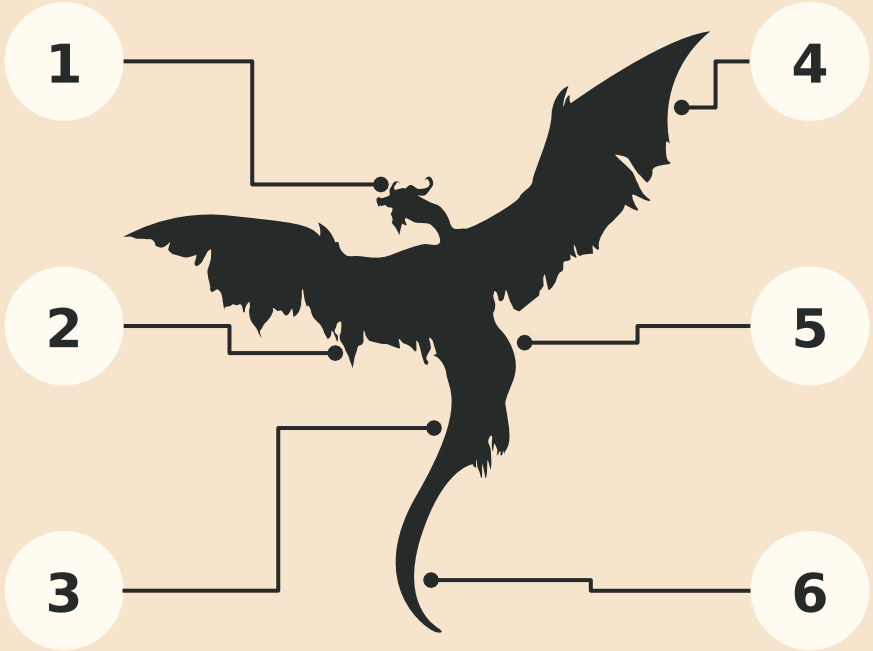
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

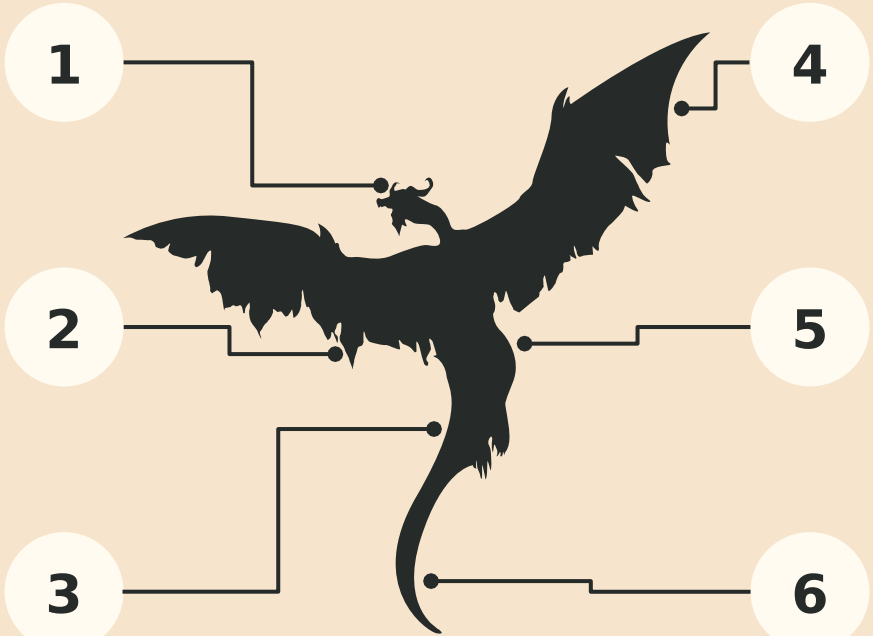
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

Generate visual report, SPDX SBOM, etc

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

Downloader

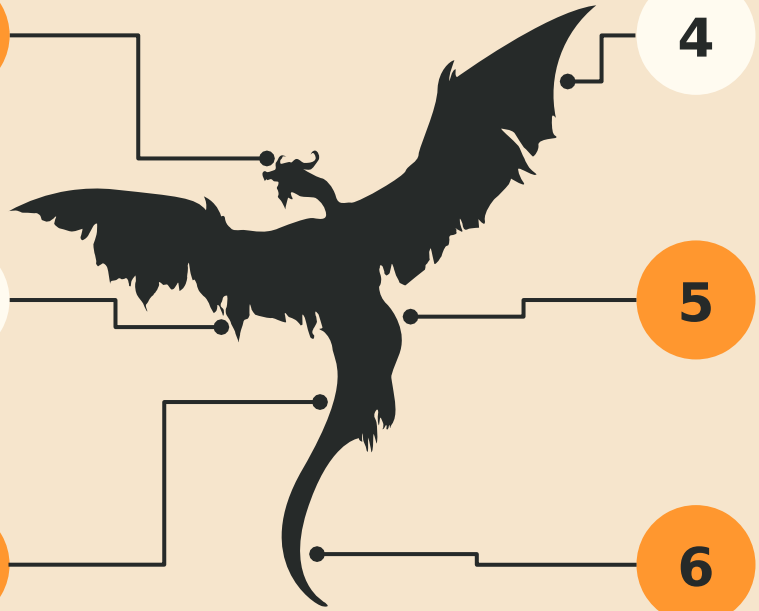
Downloads dependencies to scan

2

Scanner

Scan source code with plugin architecture

3



4

Advisor

Vulnerability scanning

5

Evaluator

Custom-based license compliance

6

Reporter

Generate visual report, SPDX SBOM, etc

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

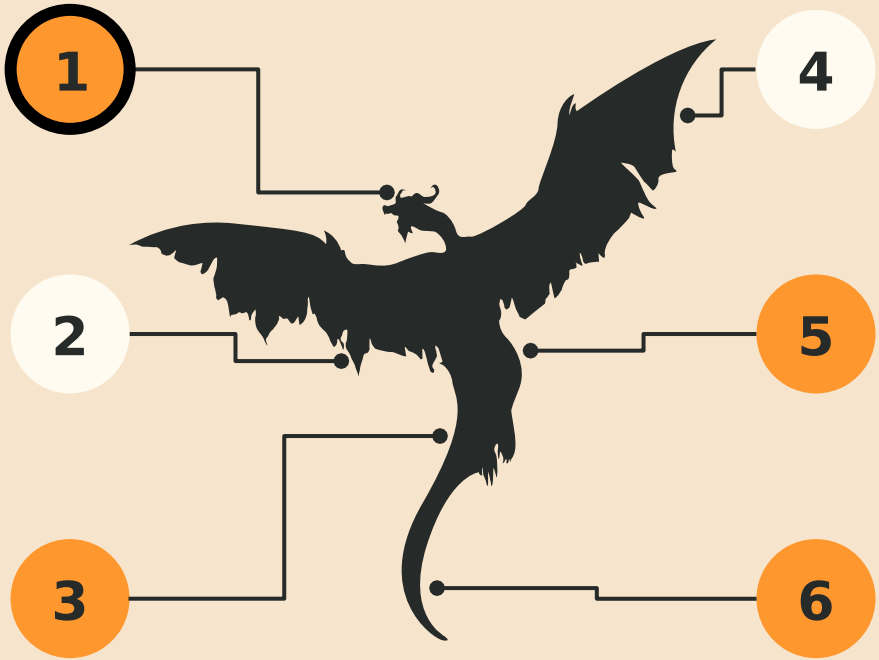
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

Generate visual report, SPDX SBOM, etc

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture

OSS Review Toolkit

Compliance & Source SBOM

Analyser

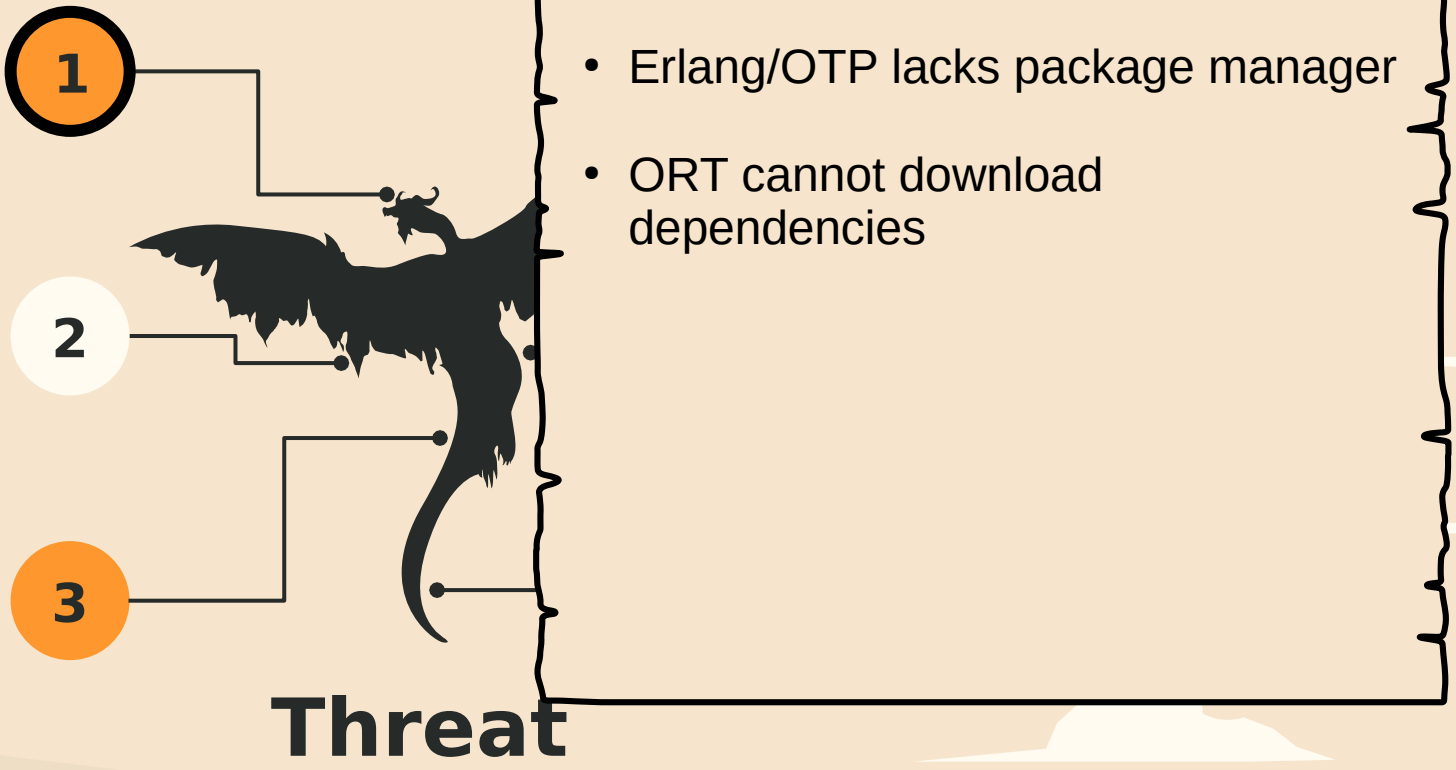
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

Downloader

Downloads dependencies to scan

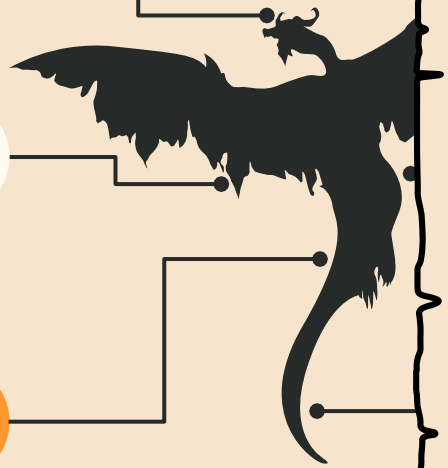
Scanner

Scan source code with plugin architecture

1

2

3



Threat

- Erlang/OTP lacks package manager
- ORT cannot download dependencies
- Our dependencies are vendor

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

Downloader

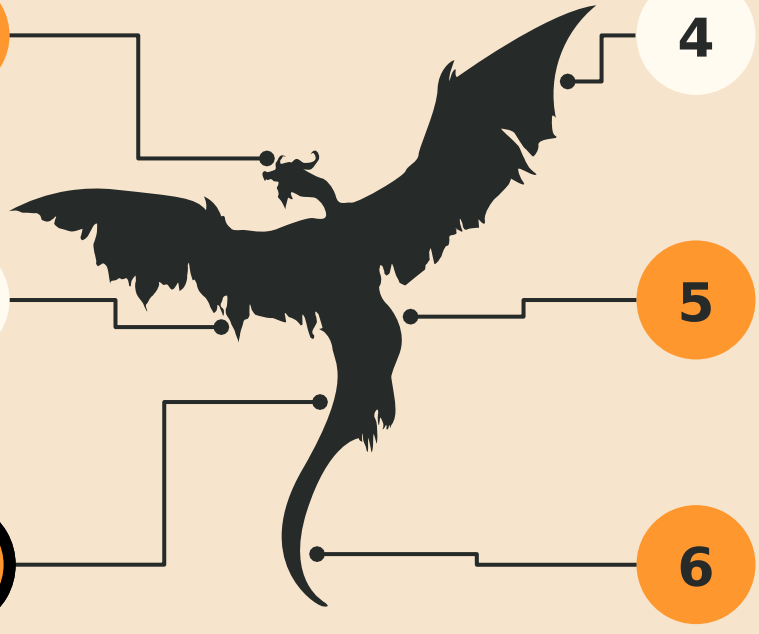
2

Downloads dependencies to scan

Scanner

3

Scan source code with plugin architecture



Advisor

4

Vulnerability scanning

Evaluator

5

Custom-based license compliance

Reporter

6

Generate visual report, SPDX SBOM, etc

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

Downloader

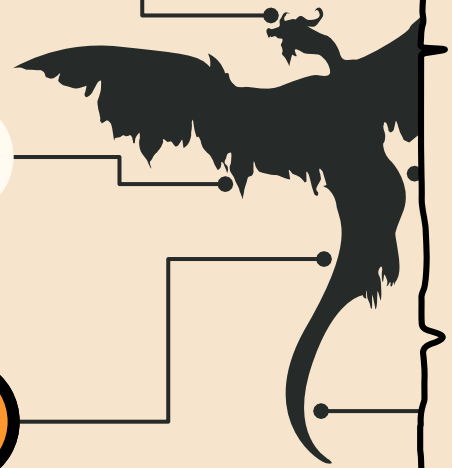
Downloads dependencies to scan

2

Scanner

Scan source code with plugin architecture

3



Threa

OSS Review Toolkit

Compliance & Source SBOM

Analyser

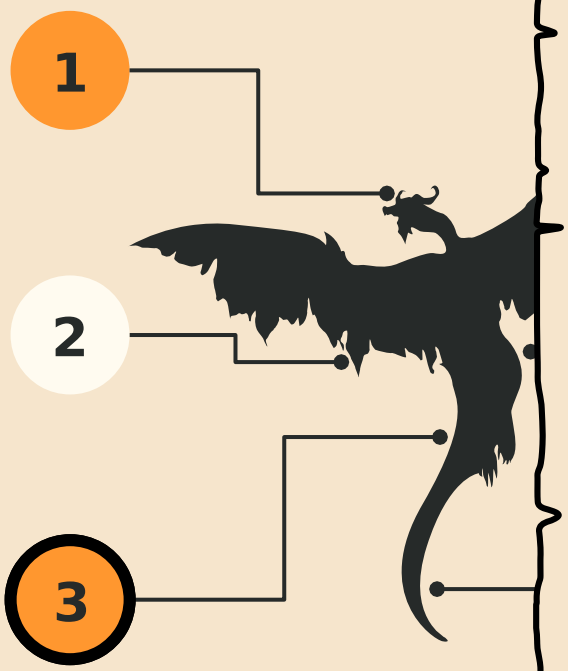
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Threa

1) Files without license are ignored

OSS Review Toolkit

Compliance & Source SBOM

Analyser

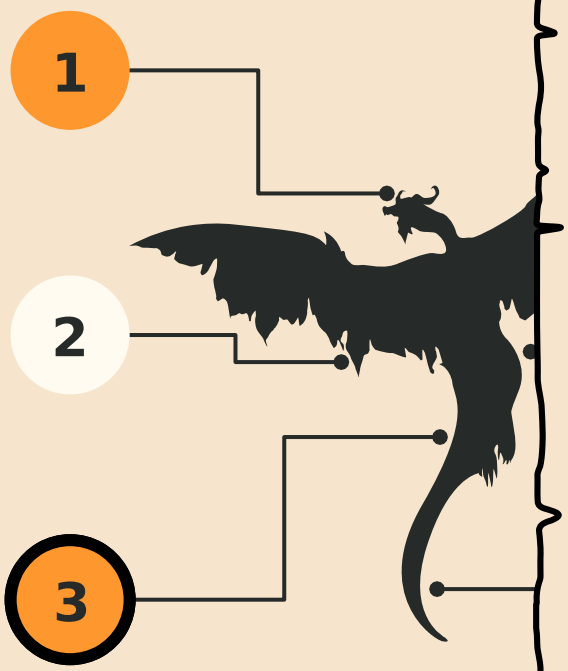
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Threa

1) Files without license are ignored

OSS Review Toolkit

Compliance & Source SBOM

Analyser

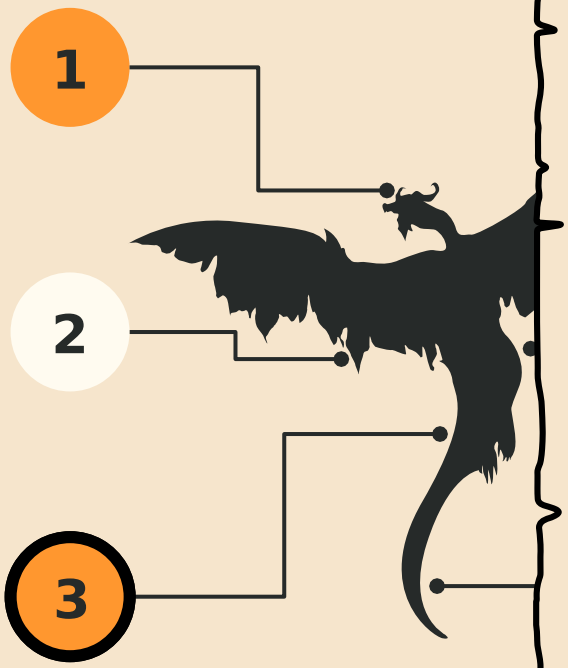
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Threa

1) Files without license are ignored

2) Erlang/OTP has 2000+ examples, tests, docs that have top-level (no) license

OSS Review Toolkit

Compliance & Source SBOM

Analyser

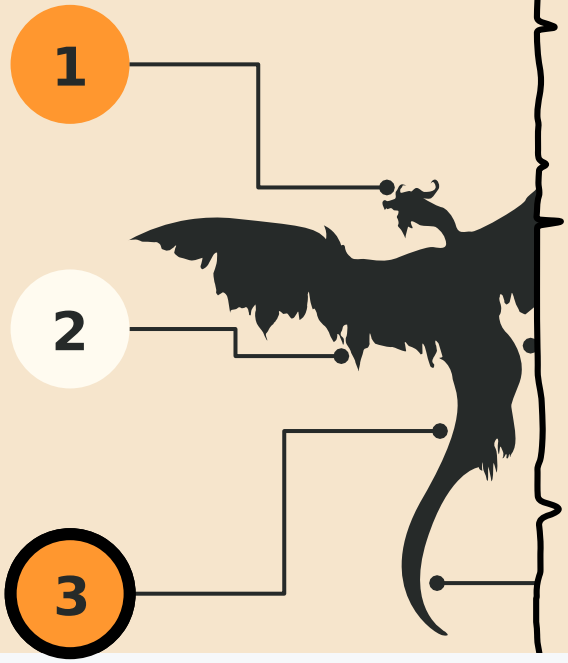
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



1) Files without license are ignored

2) Erlang/OTP has 2000+ examples, tests, docs that have top-level (no) license

- 1 and 2 via Pull request

feat(scanner): Add flag to scanner to detect unlicensed files ✓

OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

Analyser

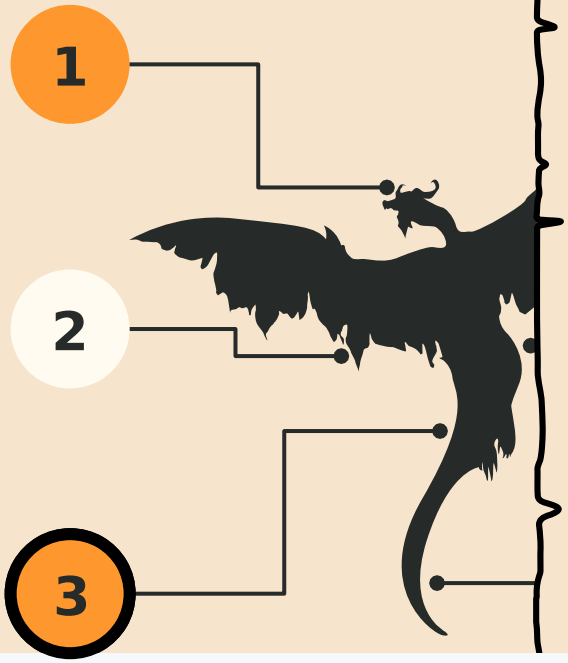
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



1) Files without license are ignored
2) Erlang/OTP has 2000+ examples, tests, docs that have top-level (no) license
3) Some licenses are wrongly reported

- 1 and 2 via Pull request

feat(scanner): Add flag to scanner to detect unlicensed files ✓

OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

Analyser

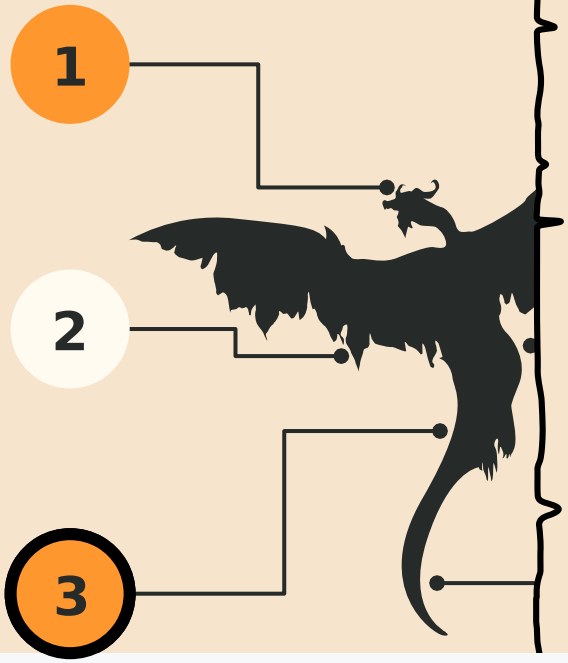
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



1) Files without license are ignored
2) Erlang/OTP has 2000+ examples, tests, docs that have top-level (no) license
3) Some licenses are wrongly reported

- 1 and 2 via Pull request
- ORT has curation facilities

feat(scanner): Add flag to scanner to detect unlicensed files ✓ OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

Analyser

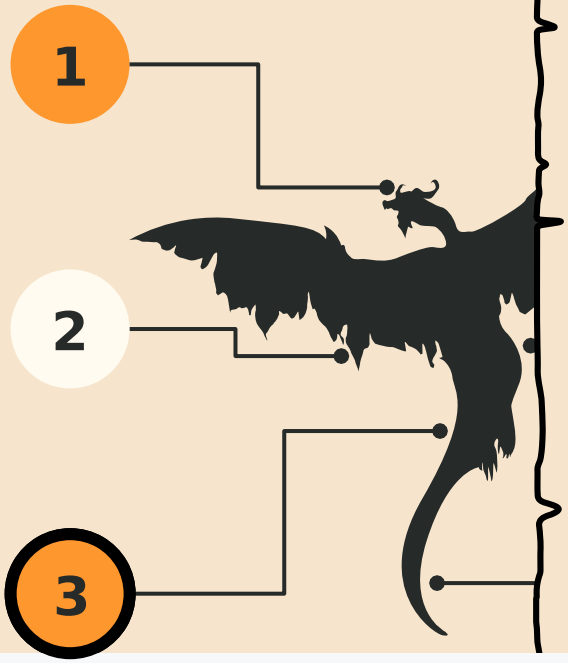
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



1) Files without license are ignored
2) Erlang/OTP has 2000+ examples, tests, docs that have top-level (no) license
3) Some licenses are wrongly reported
4) Two hours of scan time

- 1 and 2 via Pull request
- ORT has curation facilities

feat(scanner): Add flag to scanner to detect unlicensed files ✓

OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

Analyser

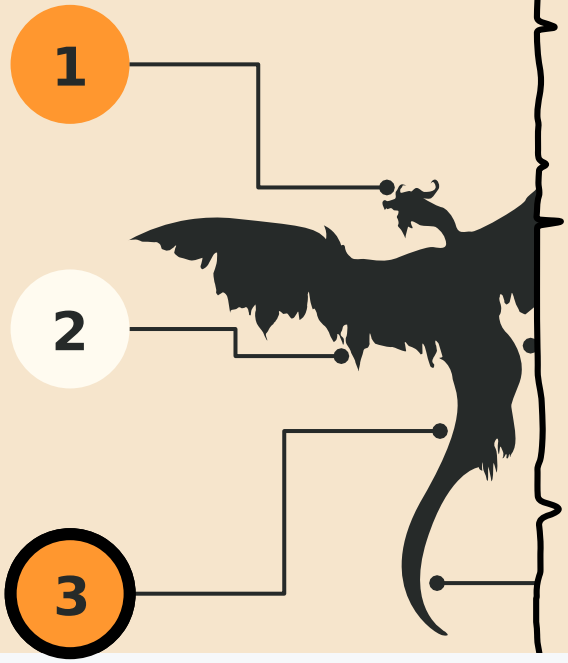
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



1) Files without license are ignored
2) Erlang/OTP has 2000+ examples, tests, docs that have top-level (no) license
3) Some licenses are wrongly reported
4) Two hours of scan time

- 1 and 2 via Pull request
- ORT has curation facilities

feat(scanner): Add flag to scanner to detect unlicensed files ✓

OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

Downloader

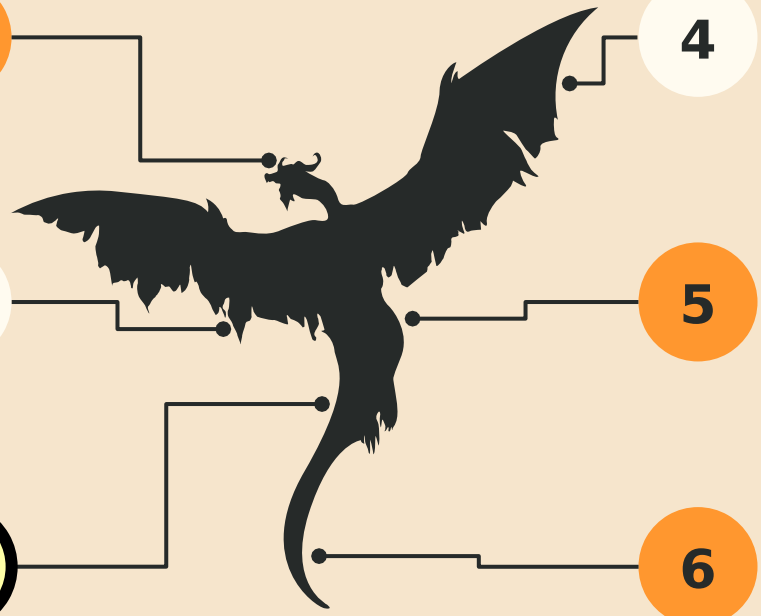
Downloads dependencies to scan

2

Scanner

Scan source code with plugin architecture

3



Advisor

Vulnerability scanning

4

Evaluator

Custom-based license compliance

5

Reporter

Generate visual report, SPDX SBOM, etc

6

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

Downloader

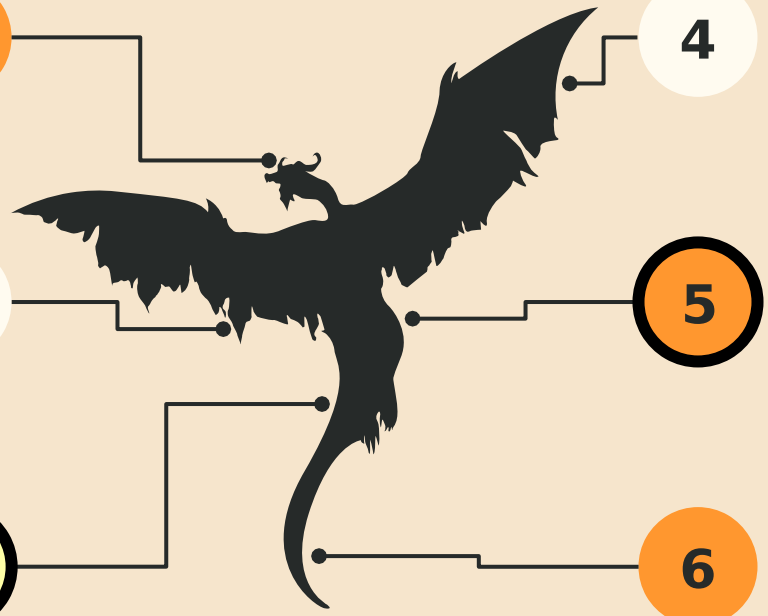
Downloads dependencies to scan

2

Scanner

Scan source code with plugin architecture

3



Advisor

Vulnerability scanning

4

Evaluator

Custom-based license compliance

5

Reporter

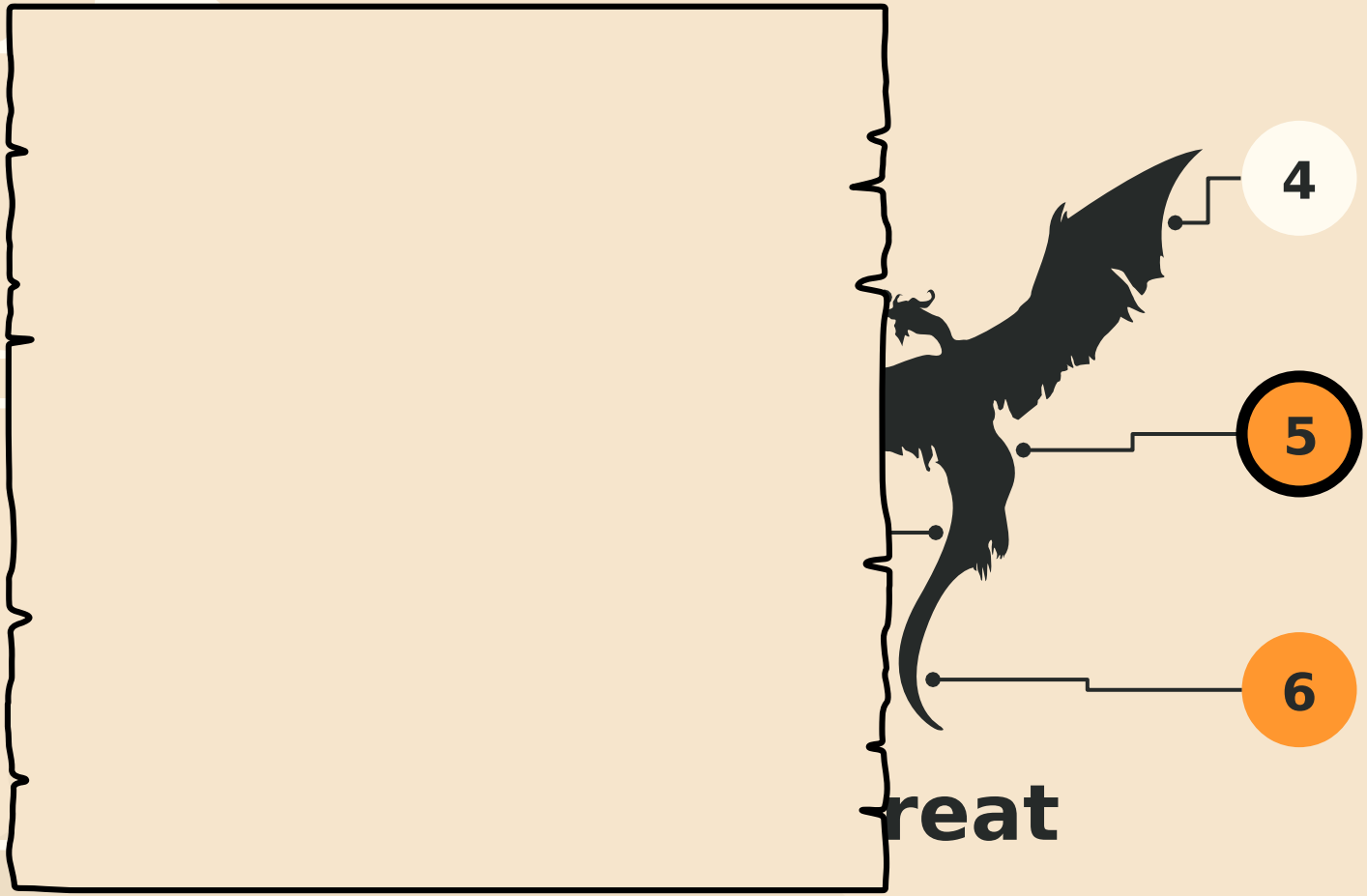
Generate visual report, SPDX SBOM, etc

6

Threat

OSS Review Toolkit

Compliance & Source SBOM



4

Advisor

Vulnerability scanning

5

Evaluator

Custom-based license compliance

6

Reporter

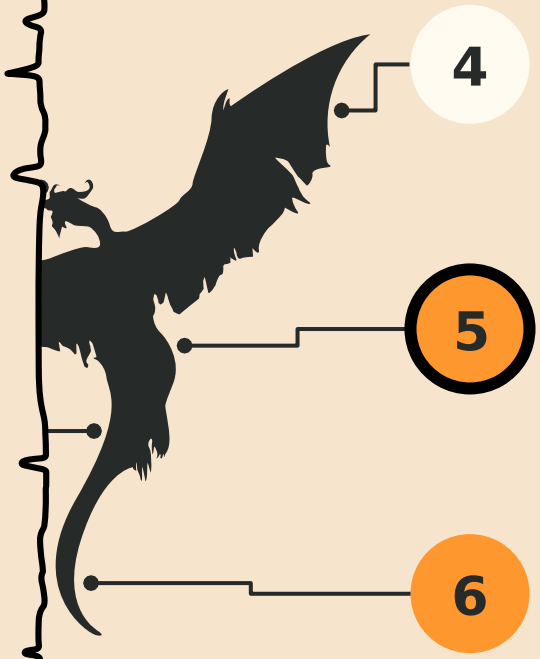
Generate visual report, SPDX SBOM, etc

reat

OSS Review Toolkit

Compliance & Source SBOM

1) Erlang/OTP lacks compliance automation



reat

Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

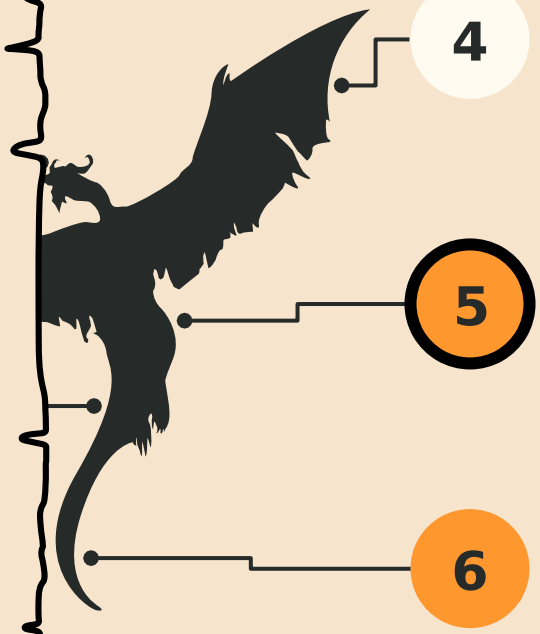
Generate visual report, SPDX SBOM, etc

OSS Review Toolkit

Compliance & Source SBOM

1) Erlang/OTP lacks compliance automation

2) Two hours scan t



reat

Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

Generate visual report, SPDX SBOM, etc

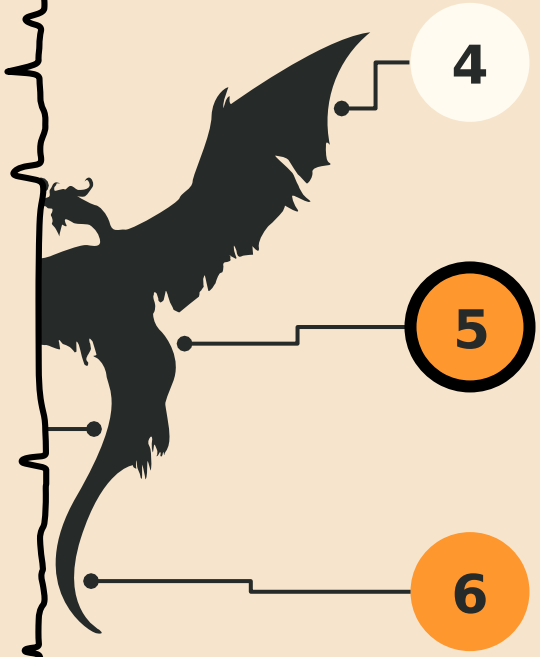
OSS Review Toolkit

Compliance & Source SBOM

1) Erlang/OTP lacks compliance automation

2) Two hours scan t

- 1 and 2 via Pull request



reat

Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

Generate visual report, SPDX SBOM, etc

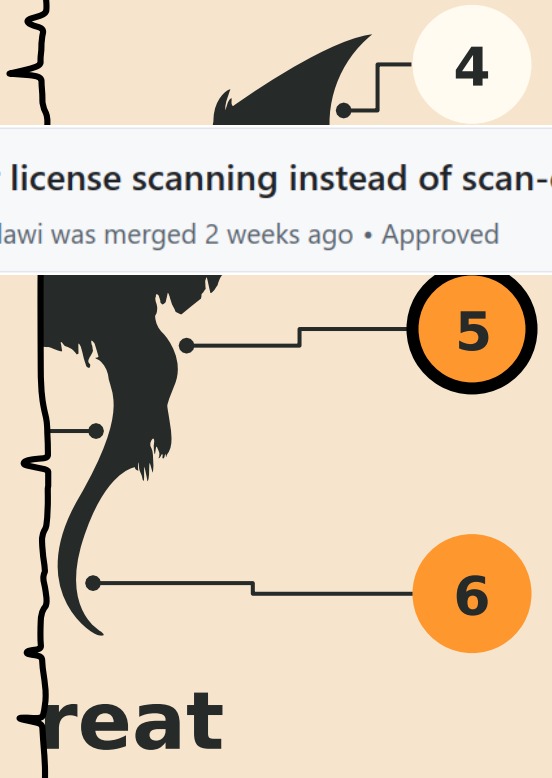
OSS Review Toolkit

Compliance & Source SBOM

1) Erlang/OTP lacks compliance automation

2) Two hours scan t

- 1 and 2 via Pull request



Use ORT for license scanning instead of scan-code ✕ enhancement team:VM

#9532 by garazdawi was merged 2 weeks ago • Approved

4

Advisor

Vulnerability scanning

5

Custom-based license compliance

6

Reporter

Generate visual report, SPDX SBOM, etc

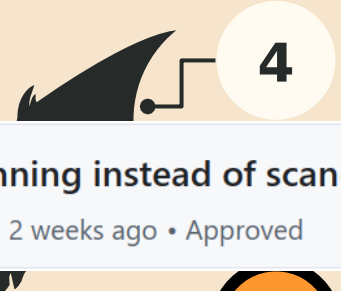
reat

OSS Review Toolkit

Compliance & Source SBOM

1) Erlang/OTP lacks compliance automation

2) Two hours scan t



Advisor

Vulnerability scanning

Use ORT for license scanning instead of scan-code ✕ enhancement team:VM

#9532 by garazdawi was merged 2 weeks ago • Approved

garazdawi commented 27 days ago — Member

Instead of relying on our own custom scan-code implementation, we rely on ort to do all analysis for us. The problem with using ort is that a re-scan of all files takes more than 1h, which is not acceptable, so this change introduces a way of scanning only the files that have changed and rely on the cache to speed up the scans.

reat

Compliance & Source SBOM

OSS Review Toolkit

1) Erlang/OTP lacks compliance automation

2) Two hours scan t

4

Advisor

Vulnerability scanning

Use ORT for license scanning instead of scan-code ✕ enhancement team:VM

#9532 by garazdawi was merged 2 weeks ago • Approved

garazdawi commented 27 days ago — Member

Instead of relying on our own custom scan-code implementation, we rely on ort to do all analysis for us. The problem with using ort is that a re-scan of all files takes more than 1h, which is not acceptable, so this change introduces a way of scanning only the files that have changed and rely on the cache to speed up the scans.

reat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

Downloader

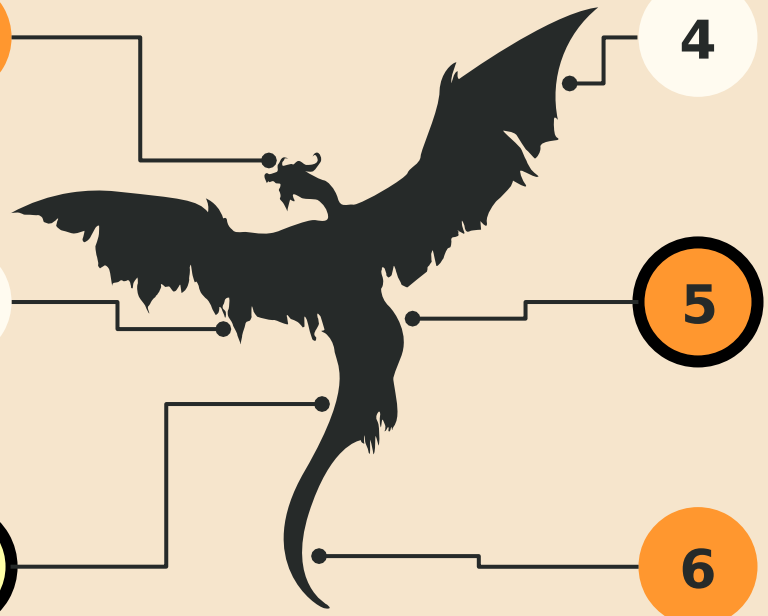
Downloads dependencies to scan

2

Scanner

Scan source code with plugin architecture

3



Advisor

Vulnerability scanning

4

Evaluator

Custom-based license compliance

5

Reporter

Generate visual report, SPDX SBOM, etc

6

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

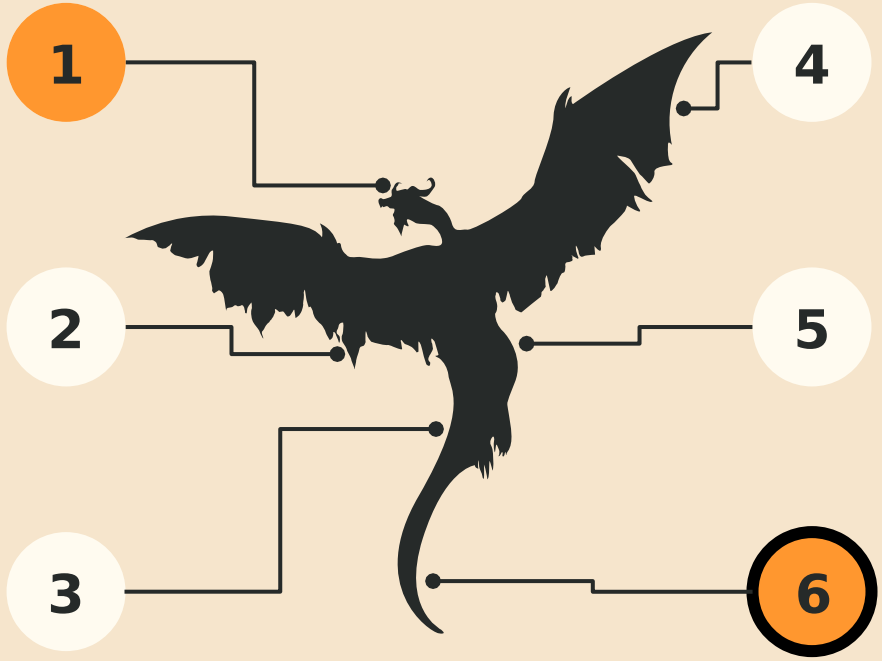
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

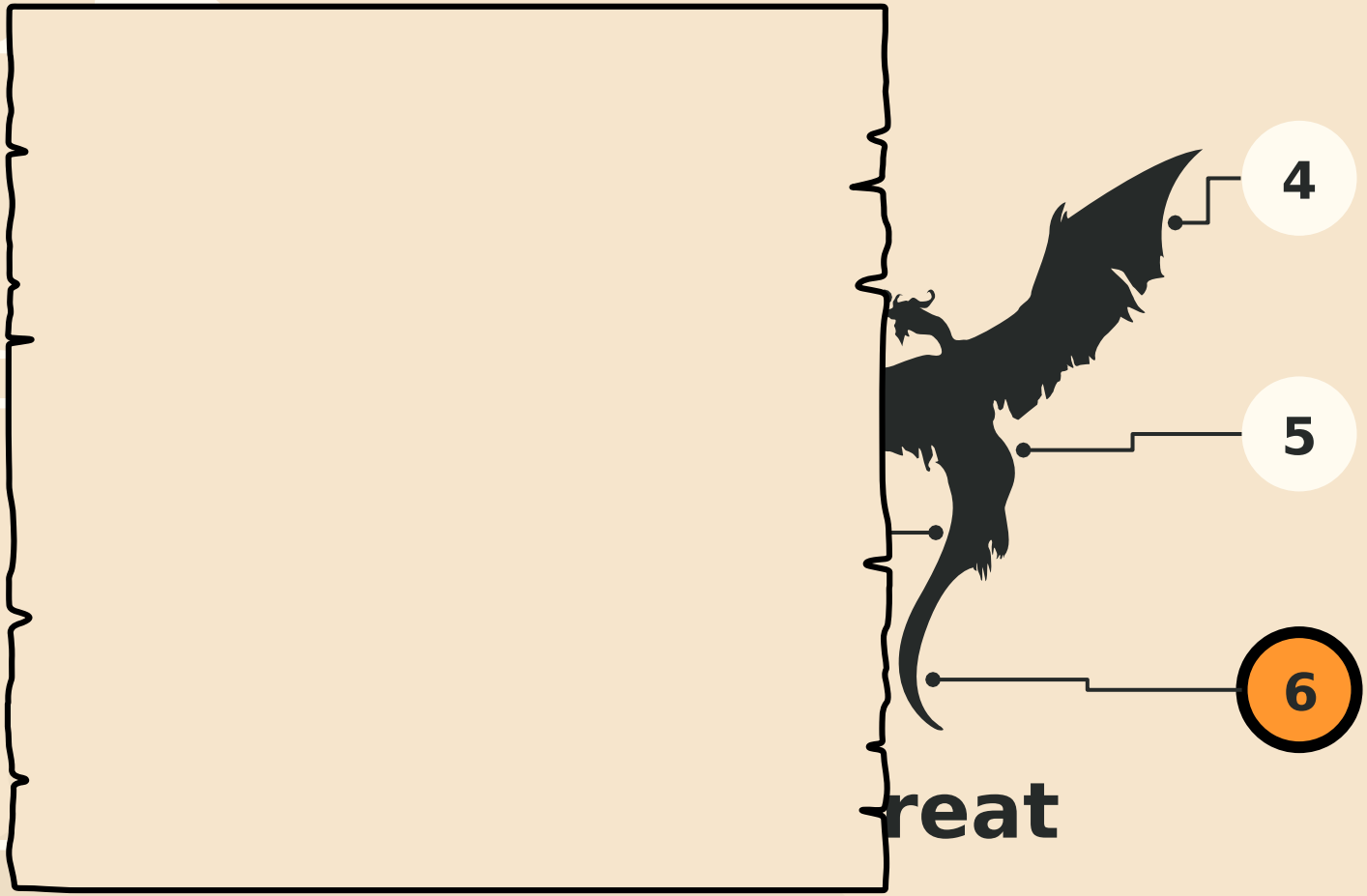
Generate visual report, SPDX SBOM, etc

Threat

OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

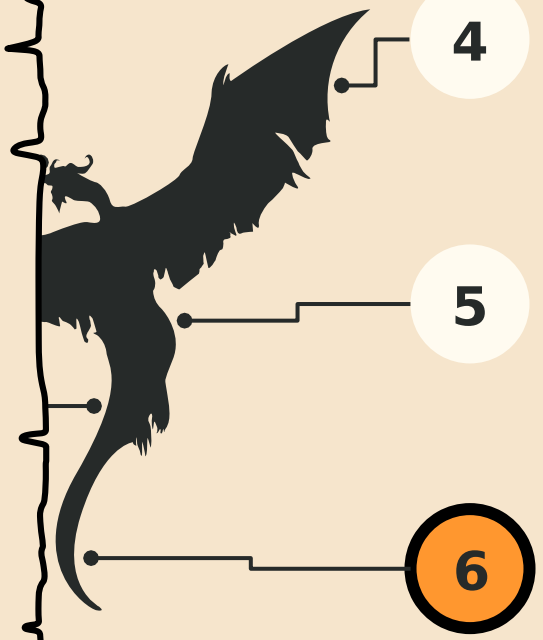
Generate visual report, SPDX SBOM, etc

OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

1) ORT does not generate source SBOM for projects without package manager



reat

Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

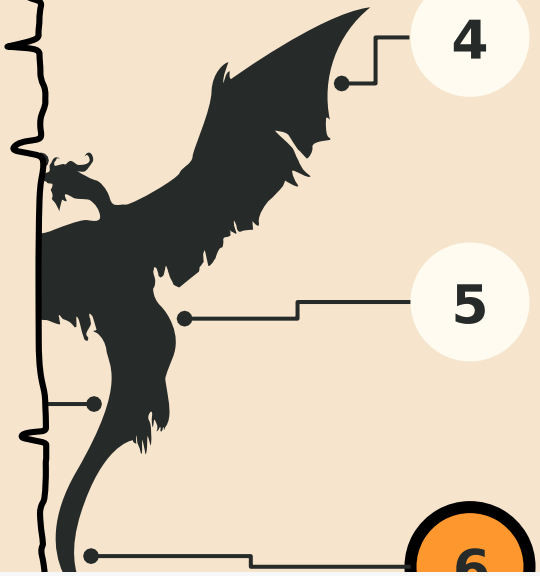
Generate visual report, SPDX SBOM, etc

OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

1) ORT does not generate source SBOM for projects without package manager



4

Advisor

Vulnerability scanning

5

Evaluator

Custom-based license compliance

6

Reporter

Generate visual report

feat(spdx): Add file level information to SPDX projects ✓

#9646 by kikofernandez was merged on Jan 23 • Approved

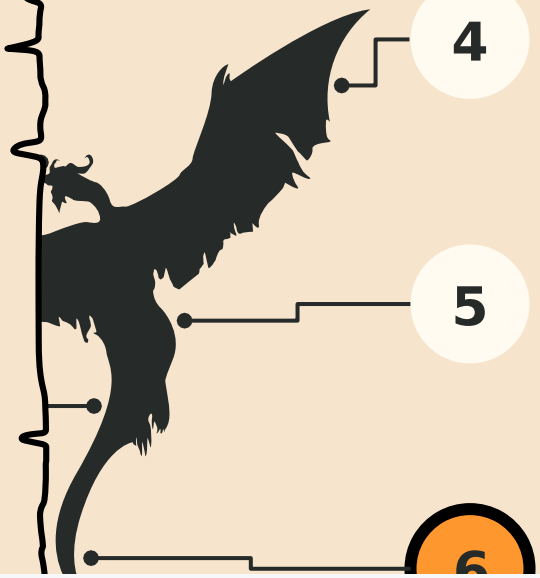
OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

1) ORT does not generate source SBOM for projects without package manager

2) ORT does not break Erlang/OTP applications on SPDX package



4

Advisor

Vulnerability scanning

5

Evaluator

Custom-based license compliance

Reporter

Generate visual report

feat(spdx): Add file level information to SPDX projects ✓

#9646 by kikofernandez was merged on Jan 23 • Approved

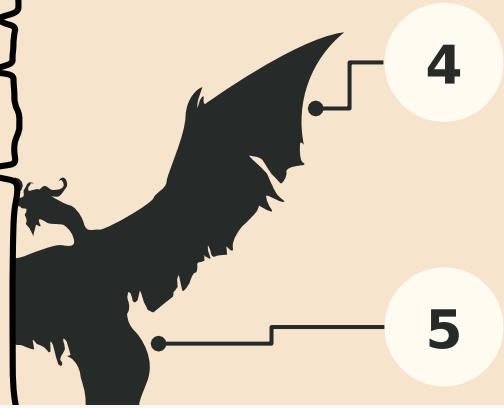
OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

1) ORT does not generate source SBOM for projects without package manager

2) ORT does not break Erlang/OTP applications on SPDX package



4

Advisor

Vulnerability scanning

5

Evaluator

Custom-based license compliance

6

Split source SBOM into multiple apps ✓ team:VM

#9586 by kikofernandez was merged 4 days ago • Approved

feat(spdx): Add file level information to SPDX projects ✓

#9646 by kikofernandez was merged on Jan 23 • Approved

OSS Review Toolkit

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

Downloader

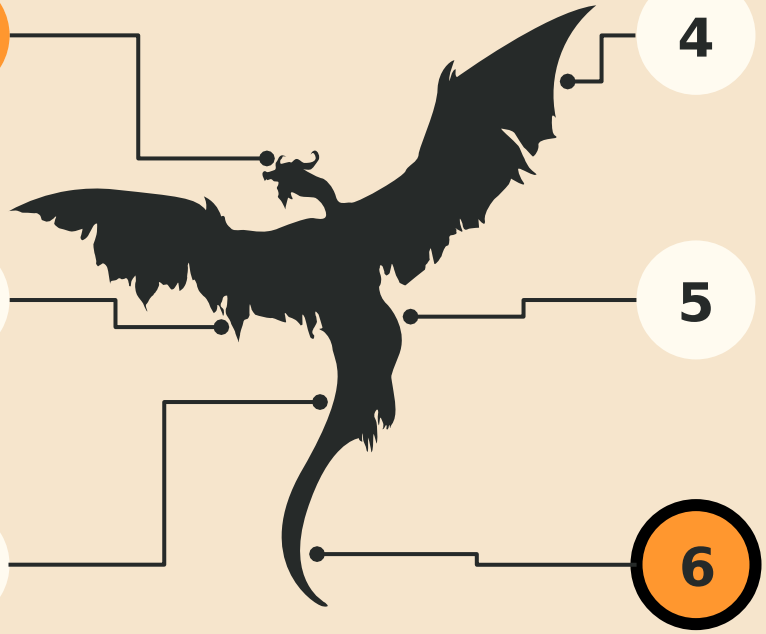
Downloads dependencies to scan

2

Scanner

Scan source code with plugin architecture

3



Threat

4

Advisor

Vulnerability scanning

5

Evaluator

Custom-based license compliance

6

Reporter

Generate visual report, SPDX SBOM, etc

OSS Review Toolkit

Compliance & Source SBOM

Analyser

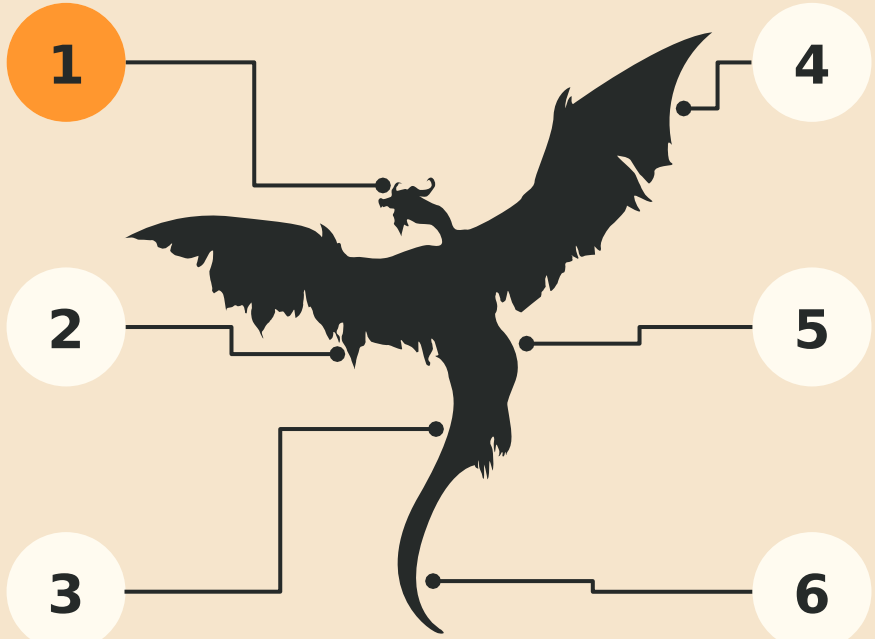
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

Generate visual report, SPDX SBOM, etc

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

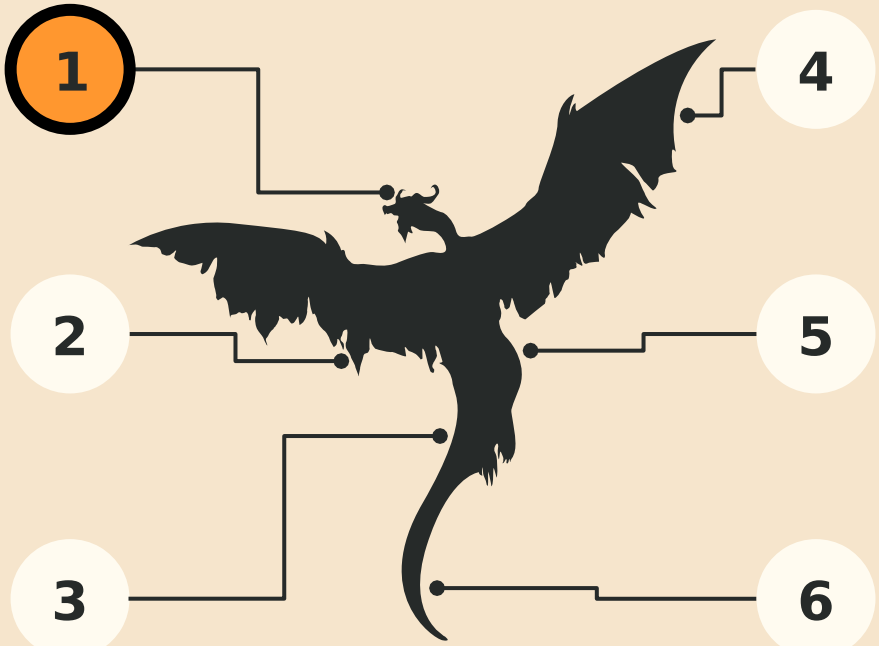
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scanner

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Reporter

Generate visual report, SPDX SBOM, etc

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

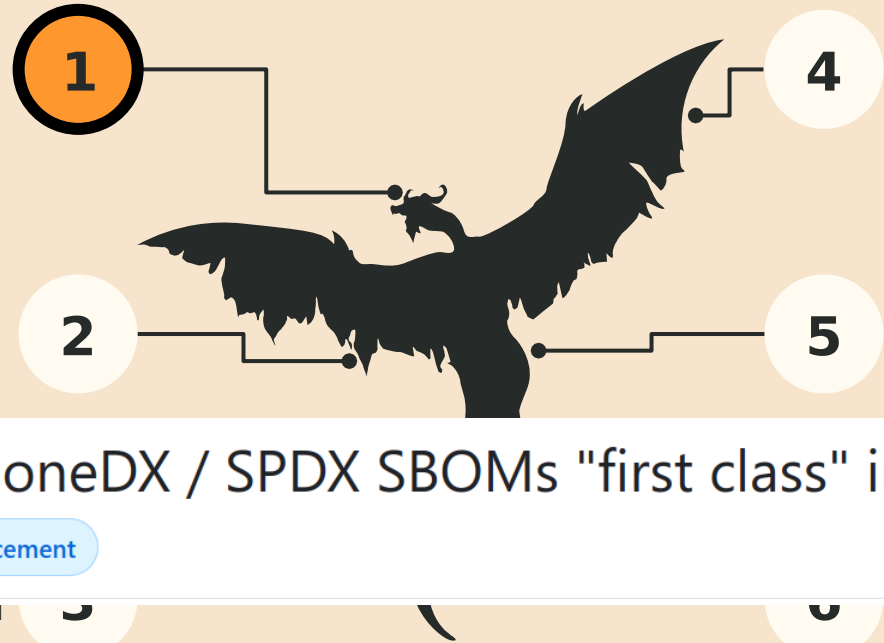
Gets dependencies of projects

Downloader

Downloads dependencies to scan

Scan

Scan source code with plugin architecture



Advisor

Vulnerability scanning

Evaluator

Custom-based license compliance

Generate visual report, SPDX SBOM, etc

Make CycloneDX / SPDX SBOMs "first class" input to ORT #9878

Open Enhancement

Threat

OSS Review Toolkit

Compliance & Source SBOM

Analyser

Gets dependencies of projects

1

Downloader

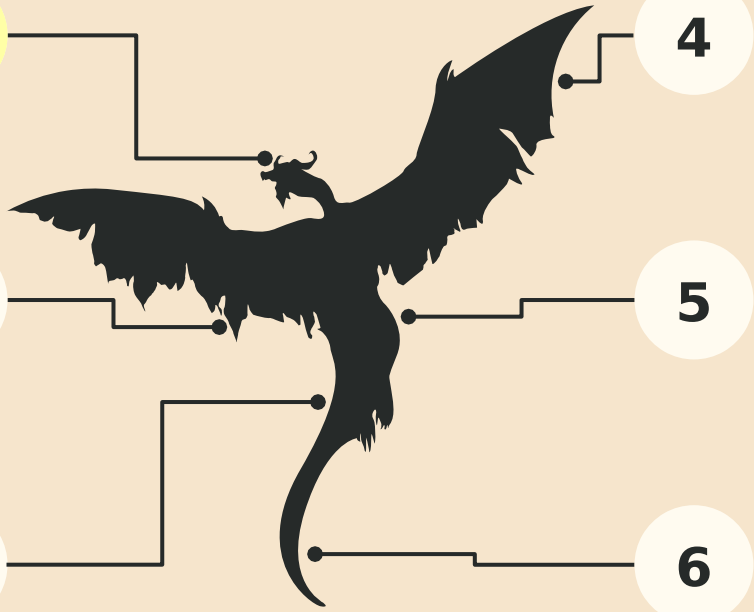
Downloads dependencies to scan

2

Scanner

Scan source code with plugin architecture

3



Advisor

Vulnerability scanning

4

Evaluator

Custom-based license compliance

5

Reporter

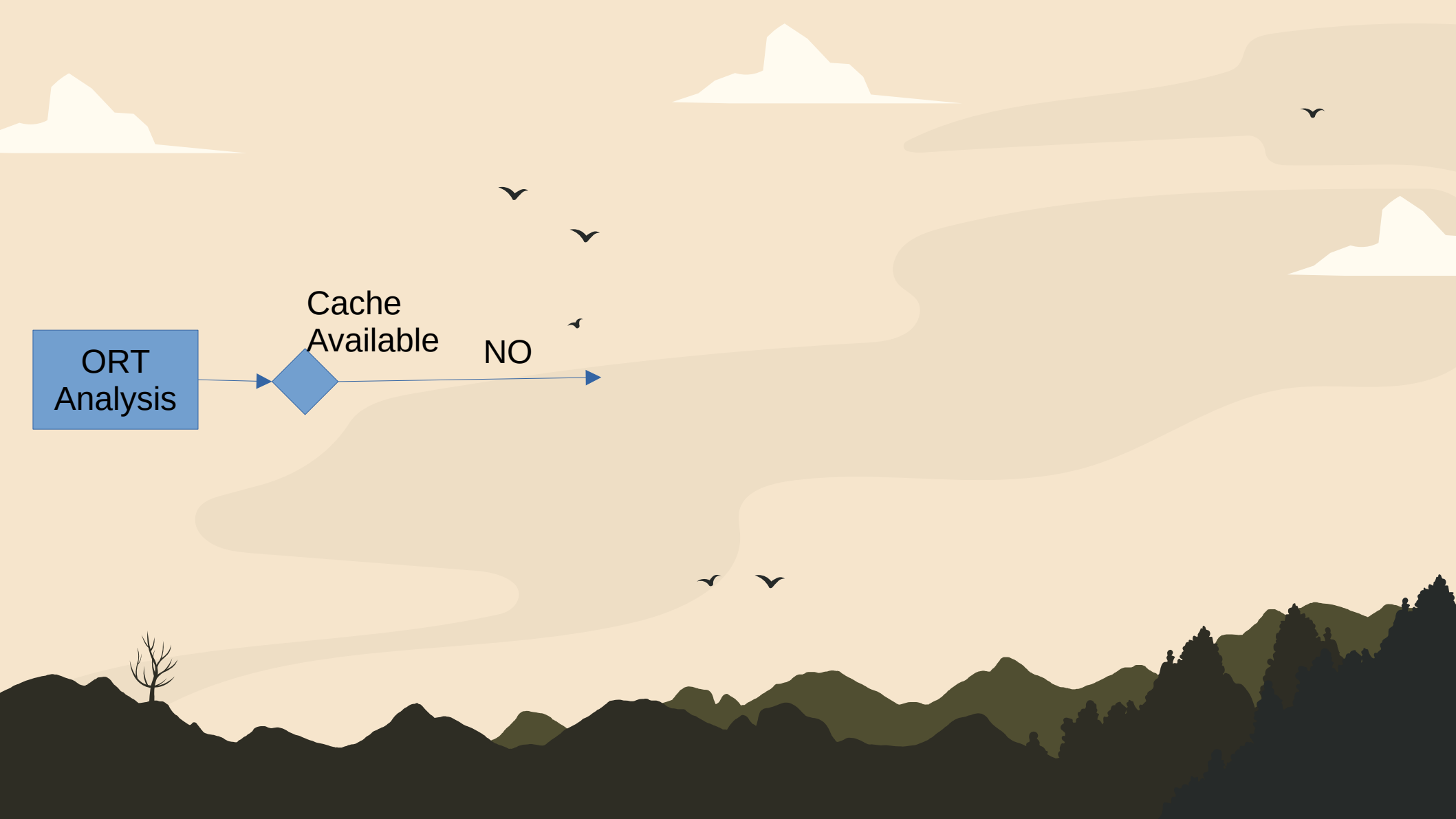
Generate visual report, SPDX SBOM, etc

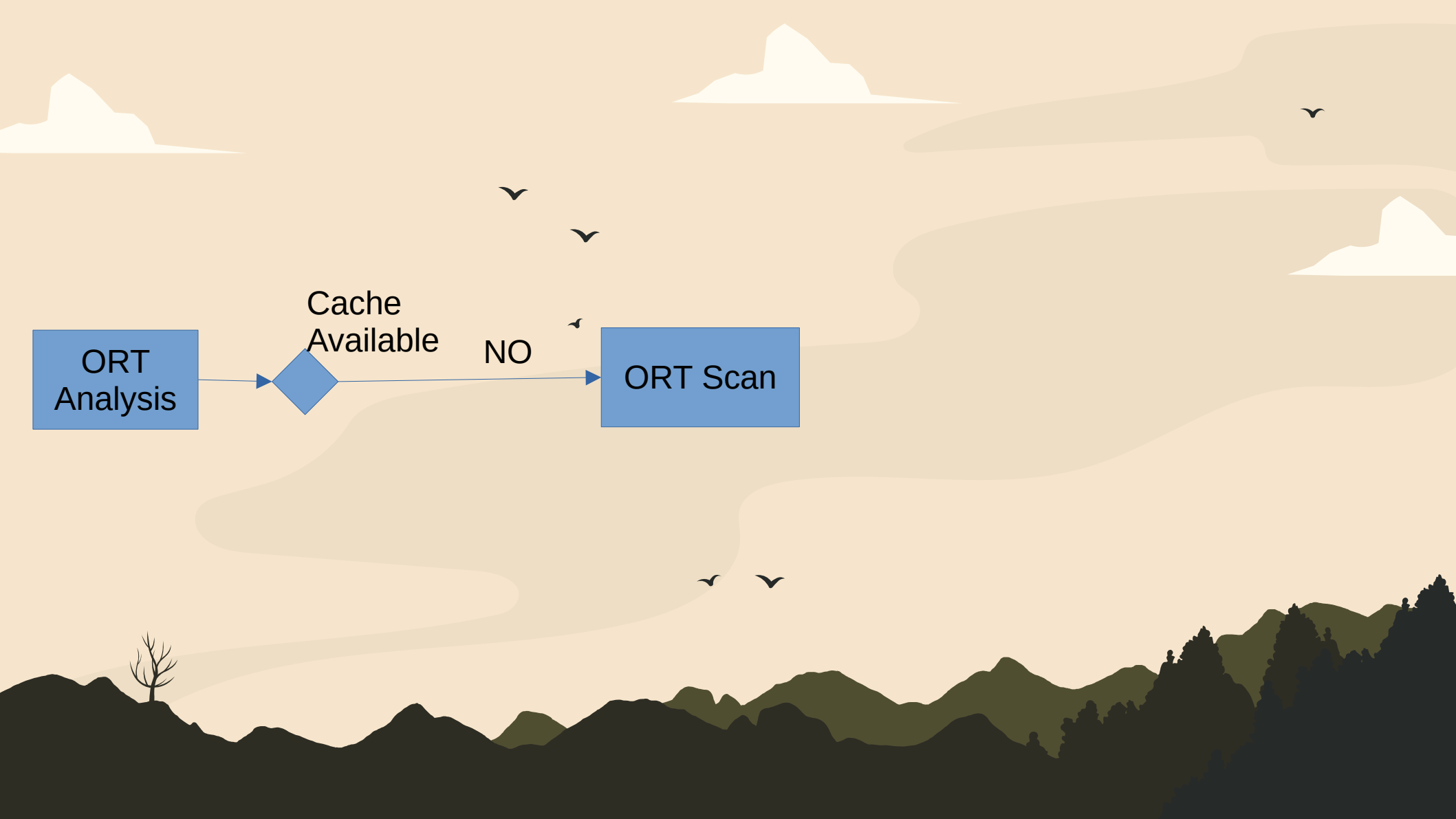
6

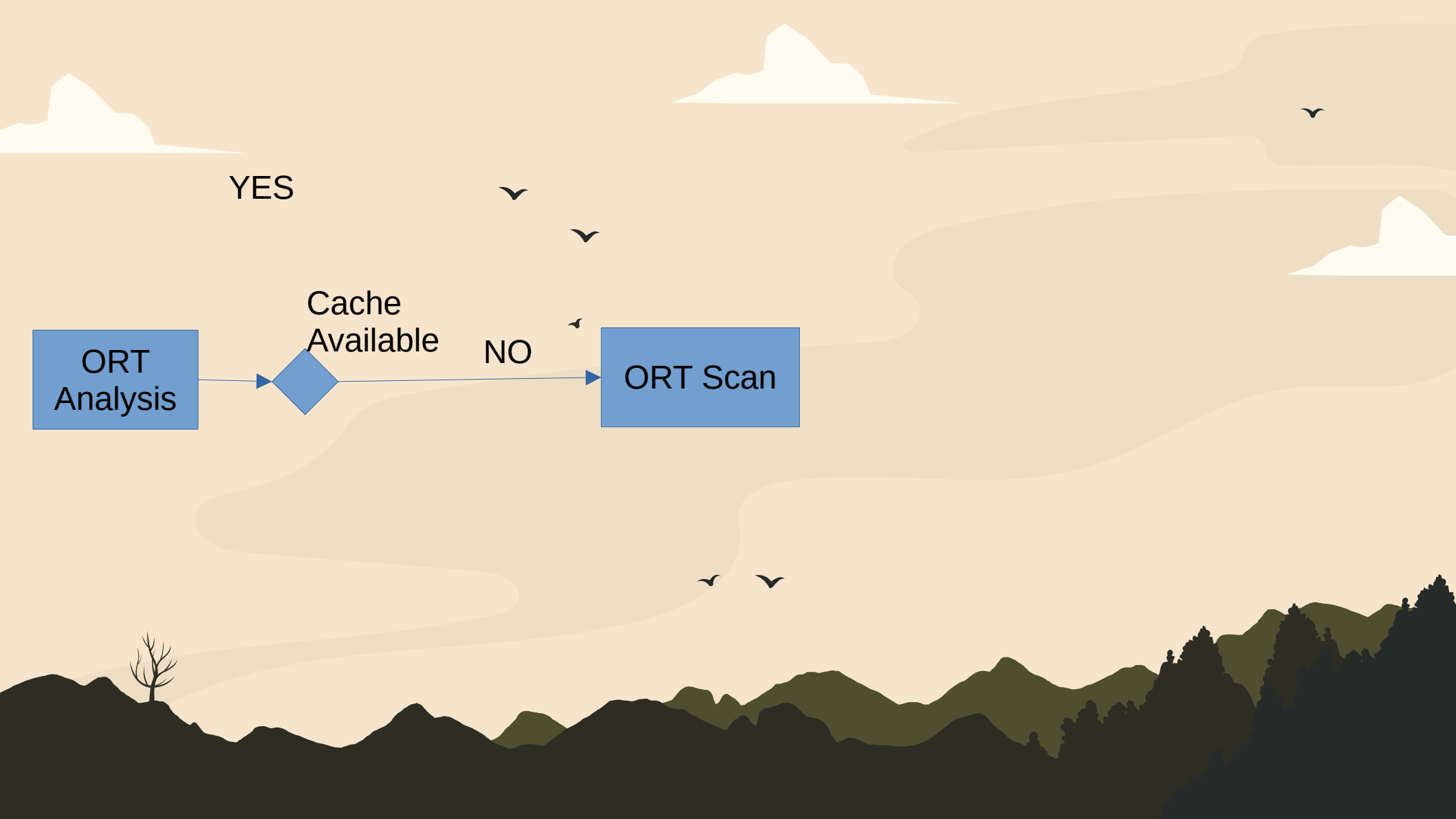
Threat

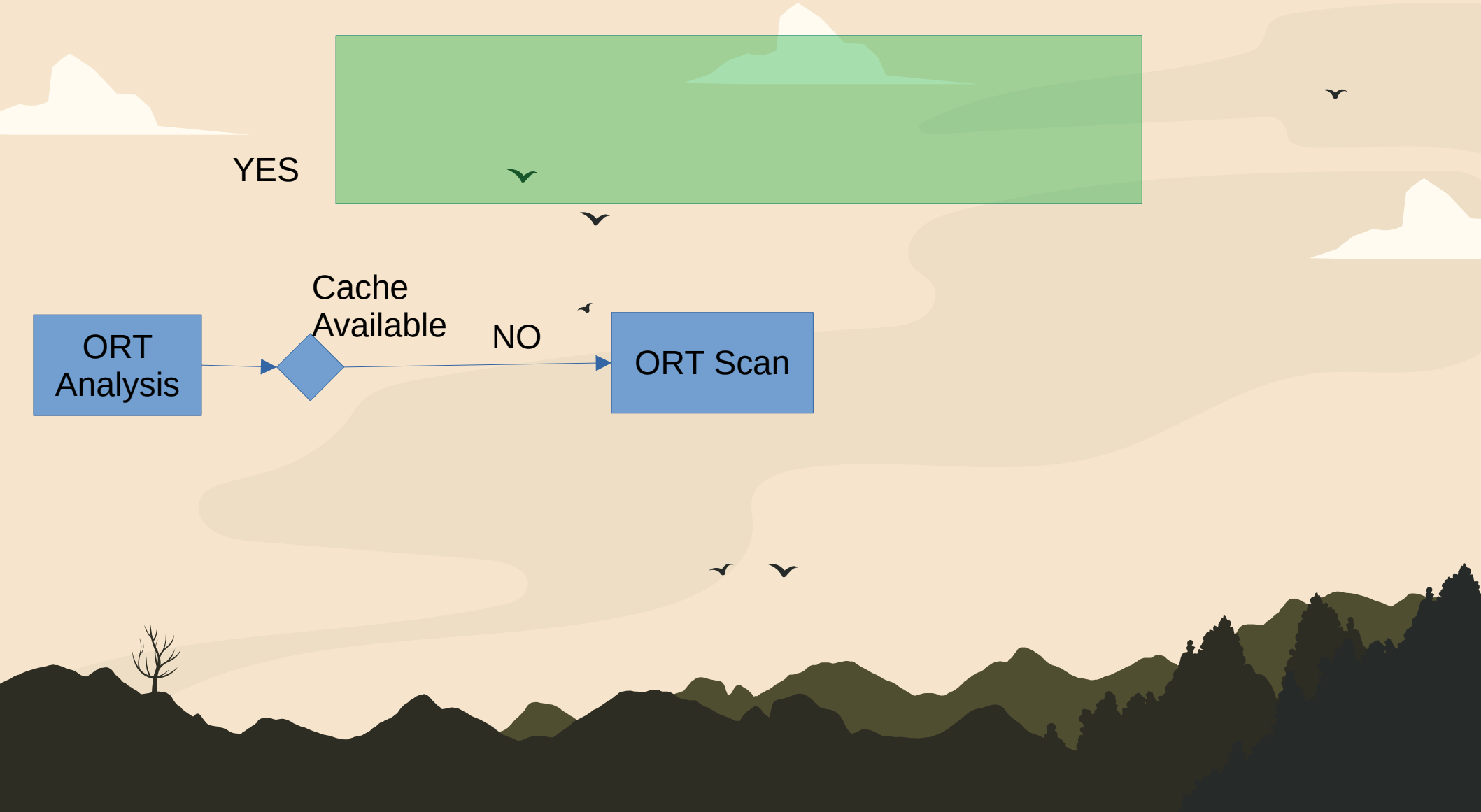

ORT
Analysis

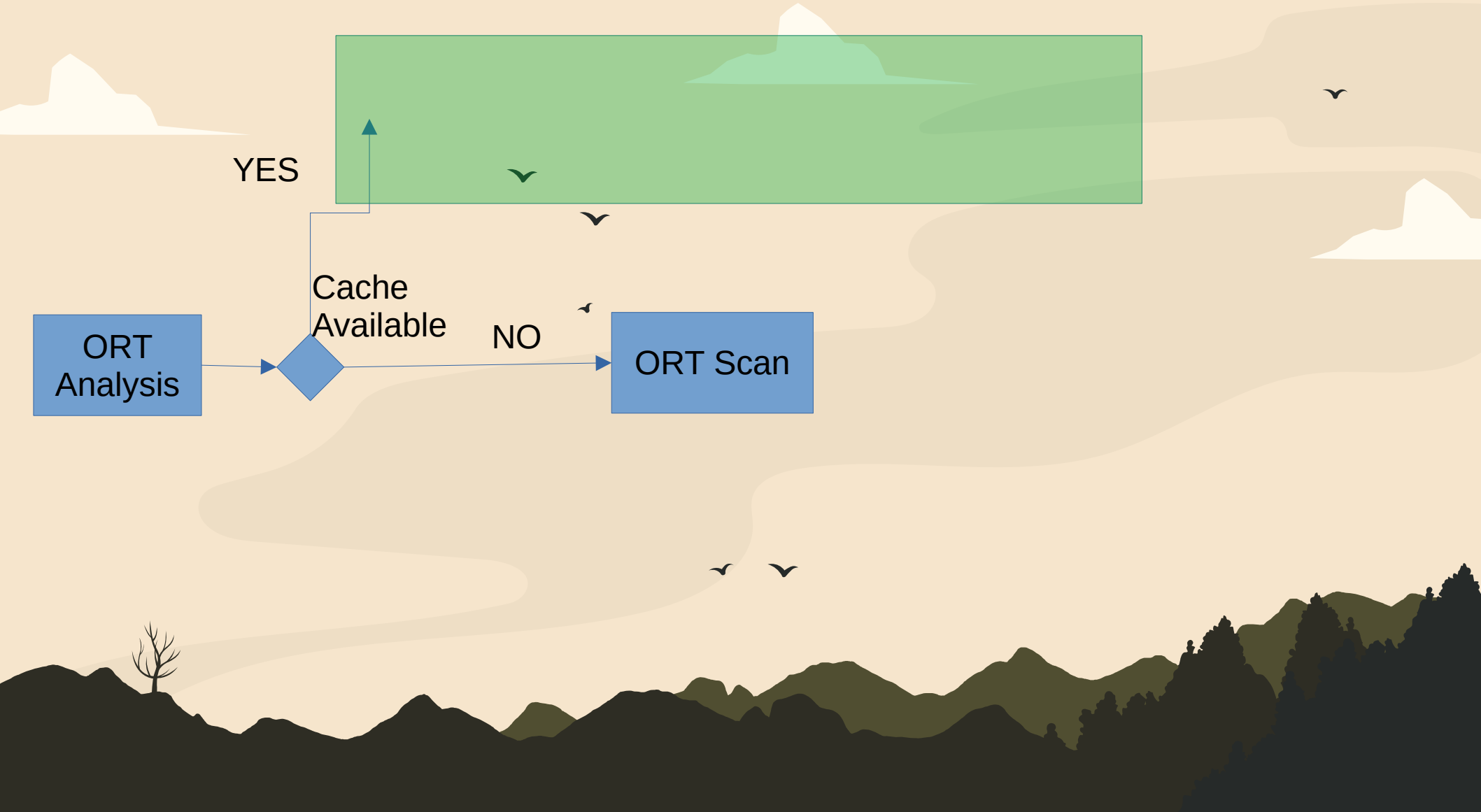

ORT Analysis
Cache Available

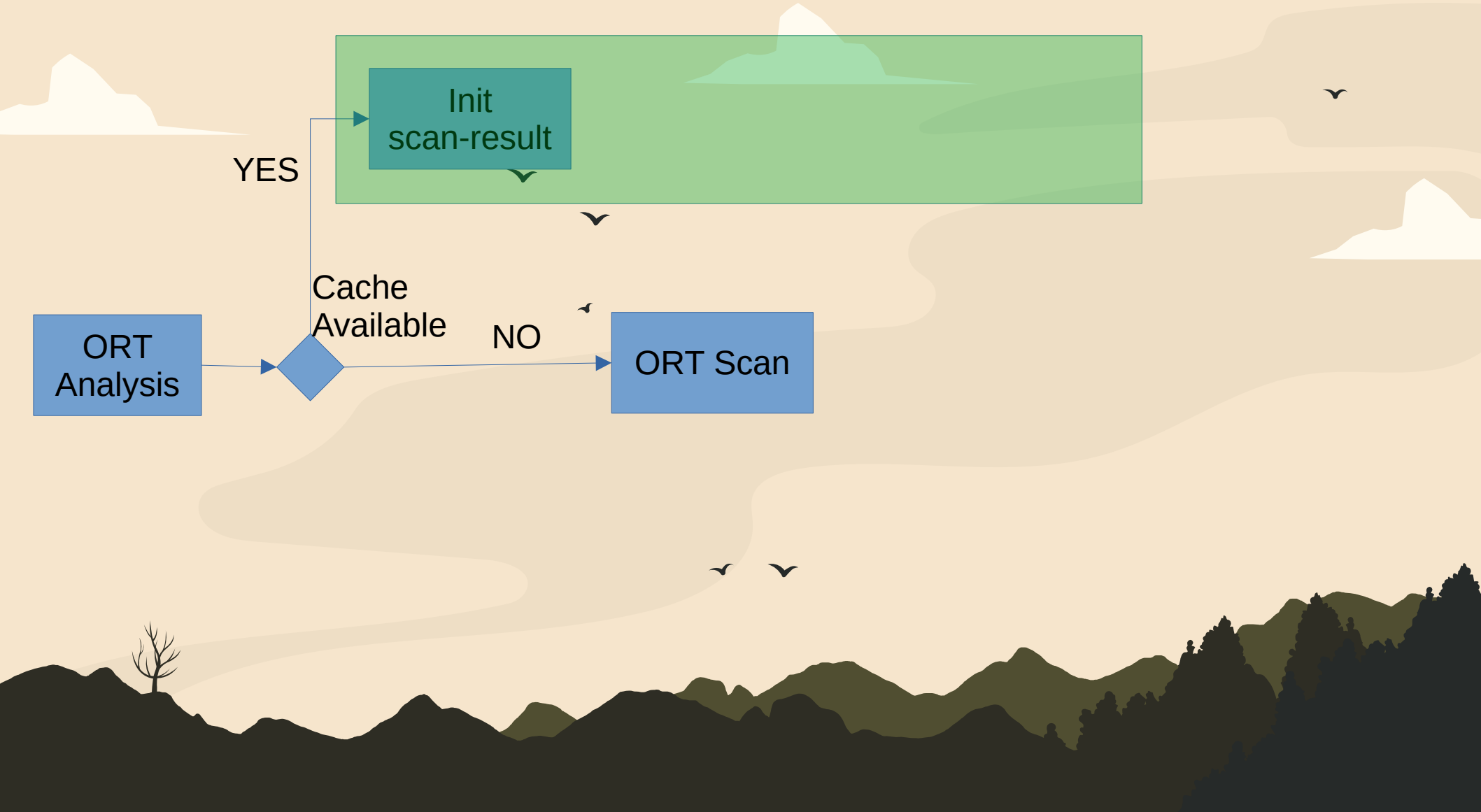

ORT Analysis
Cache Available
NO

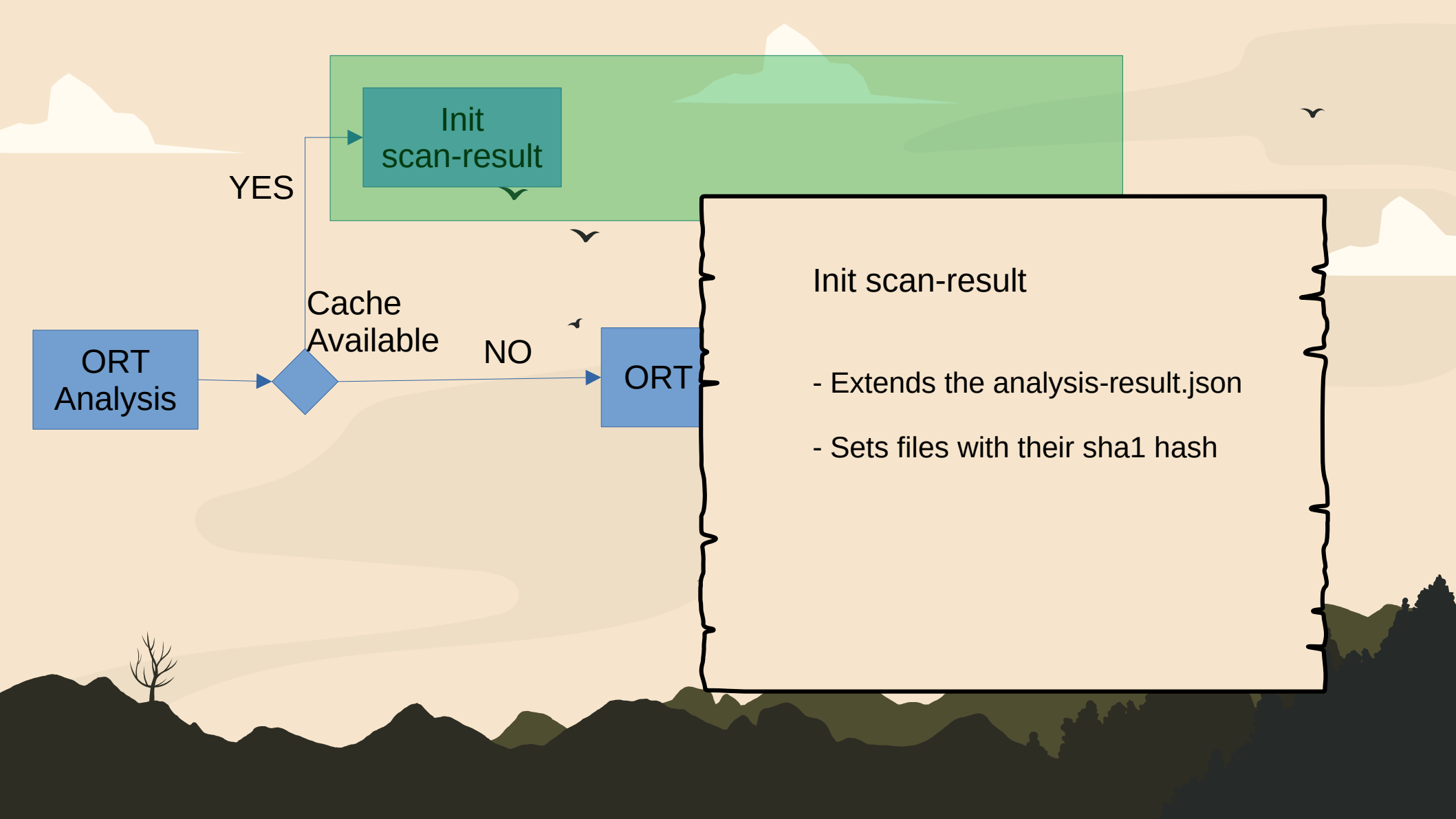

ORT Analysis
Cache Available
NO
ORT Scan

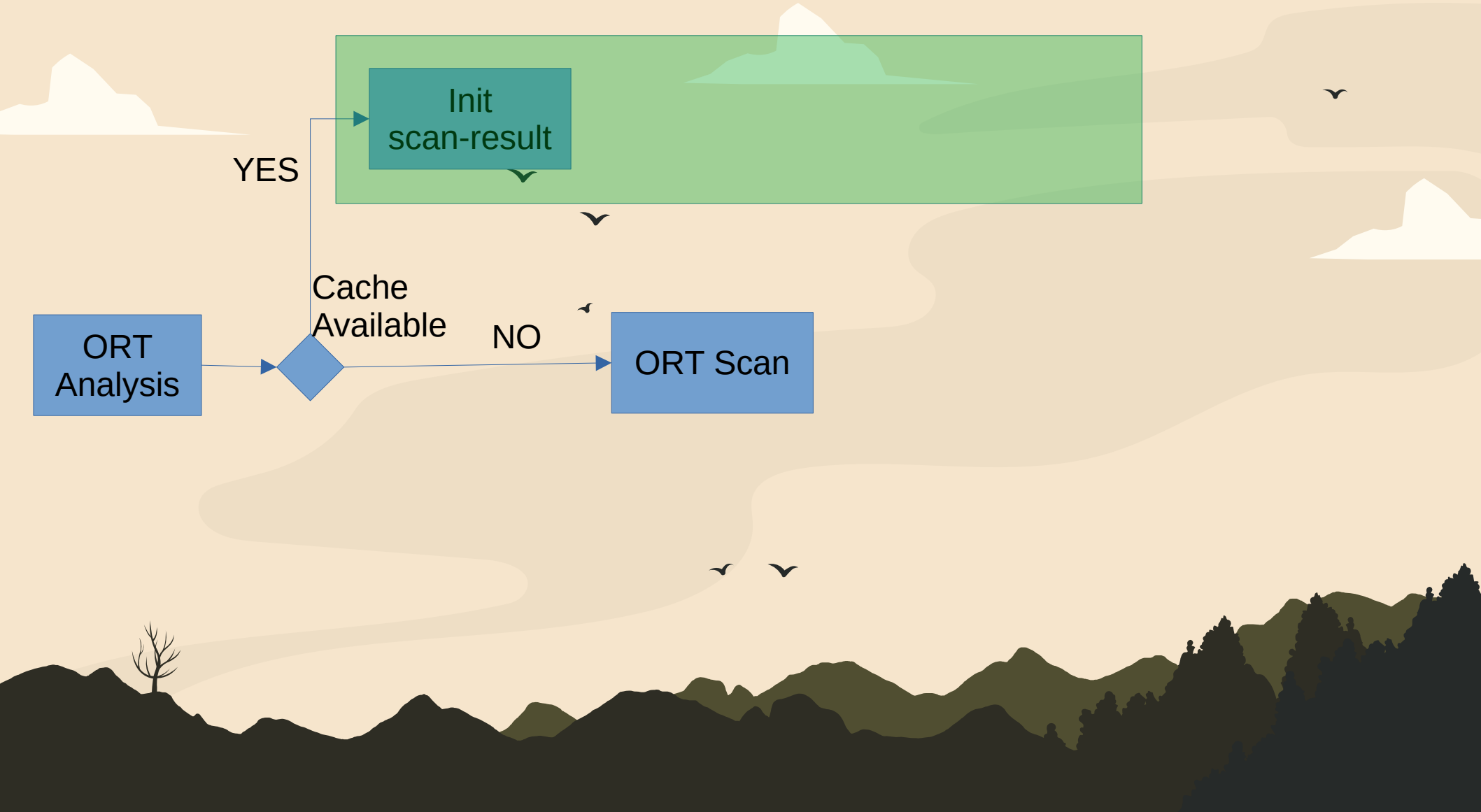

YES
Cache Available
NO
ORT Analysis
ORT Scan

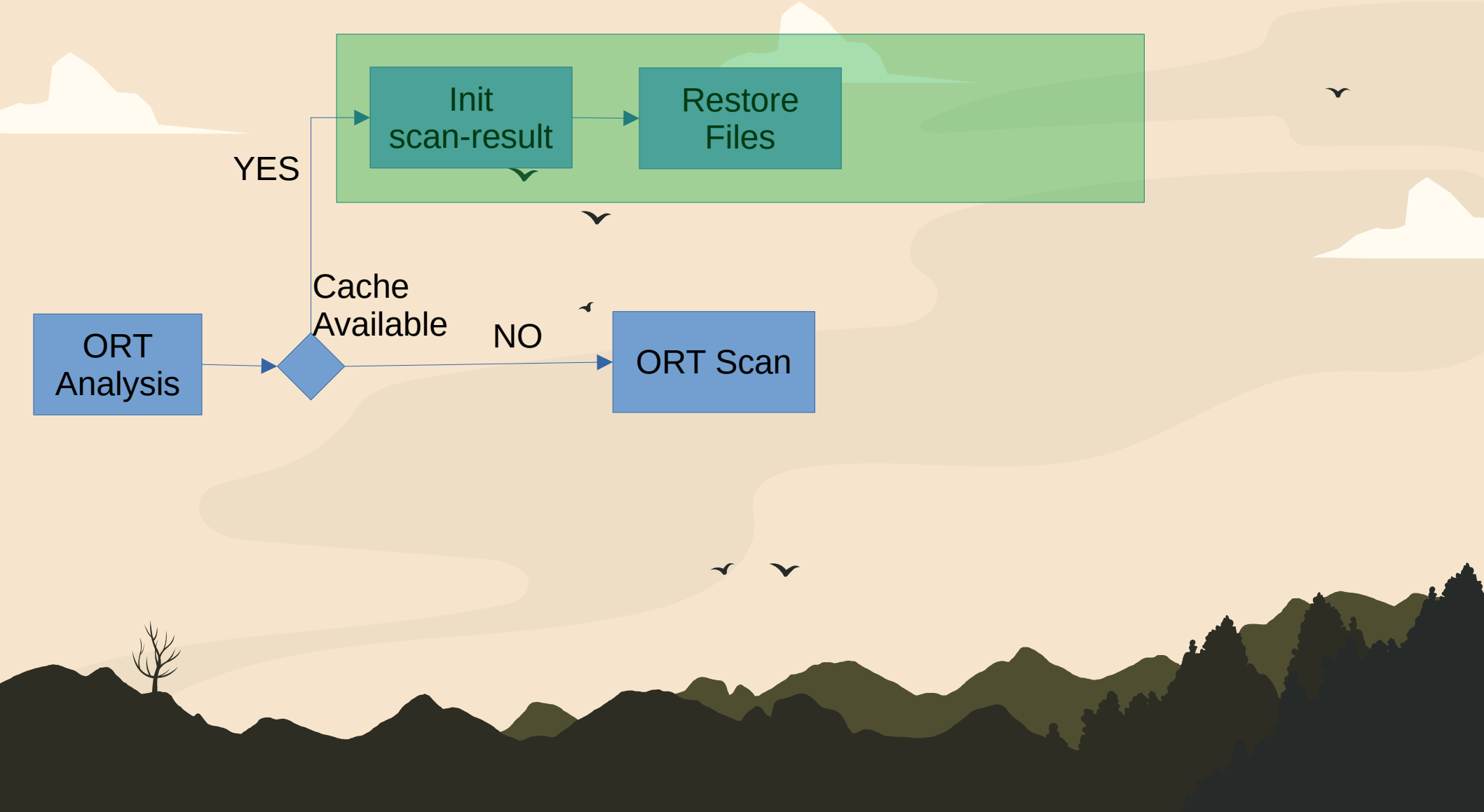

YES
Cache
Available
NO
ORT
Analysis
ORT Scan

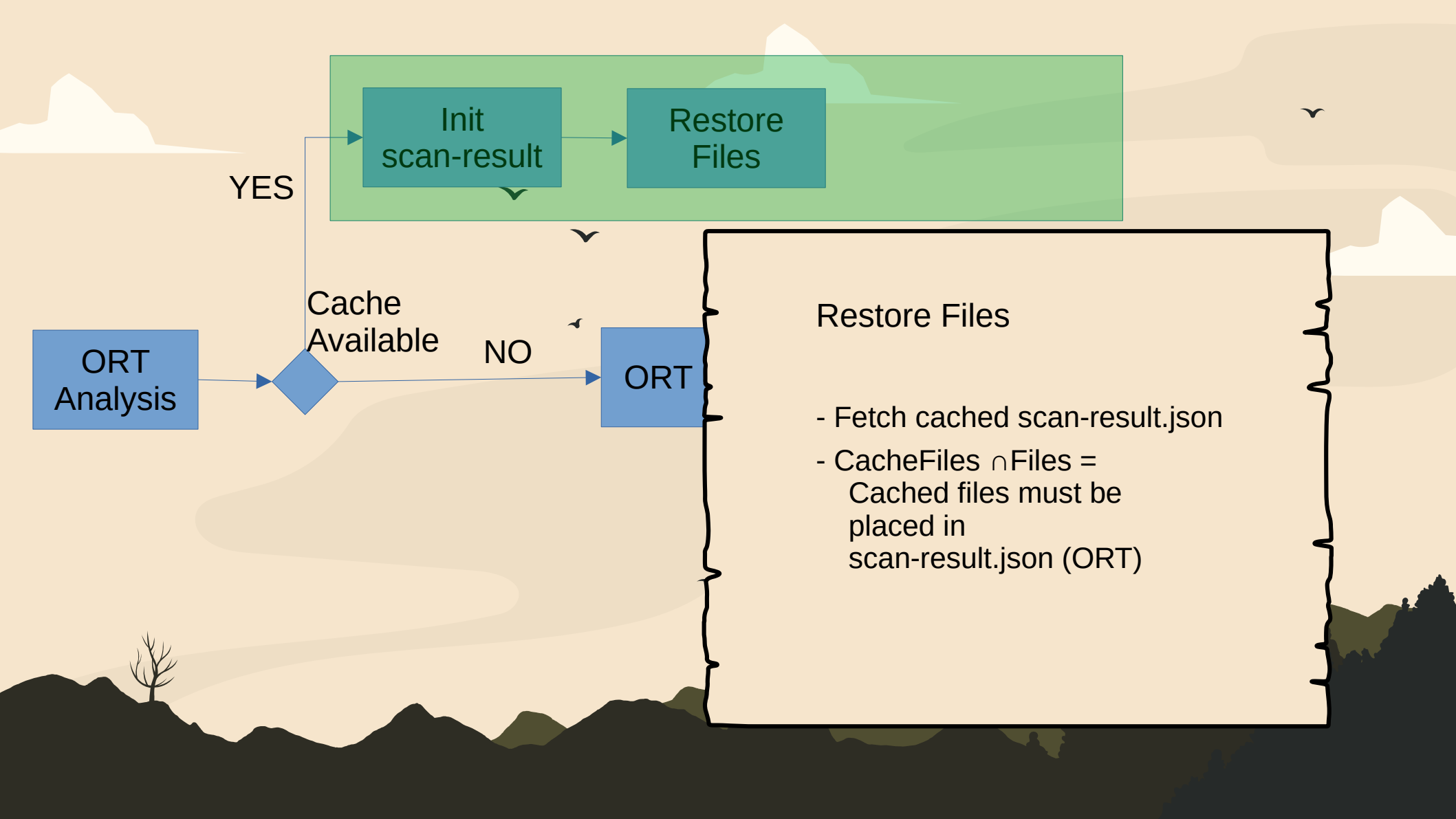

YES
Cache
Available
NO
ORT
Analysis
ORT Scan

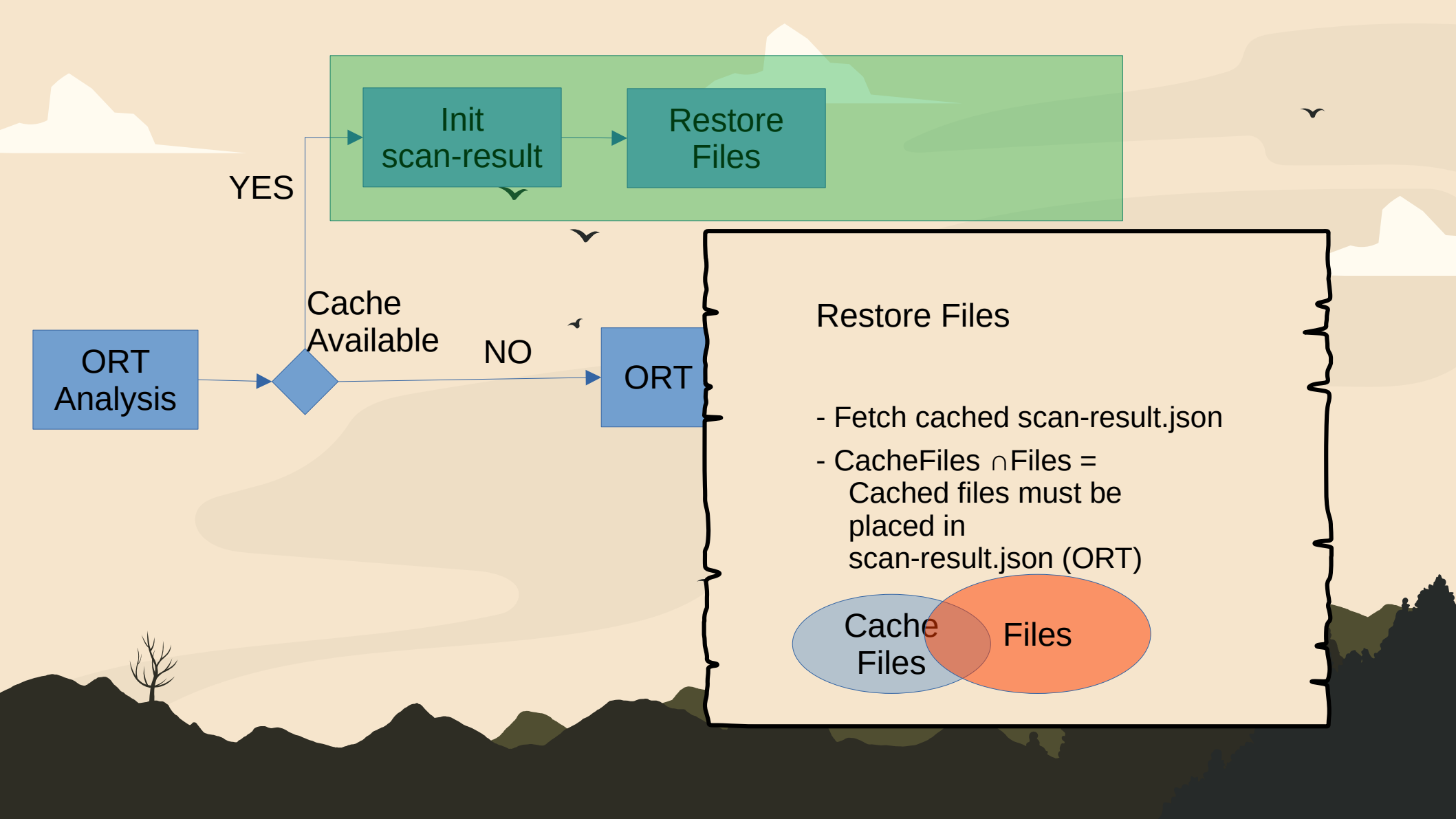

Init
scan-result
YES
Cache
Available
ORT
Analysis
NO
ORT Scan

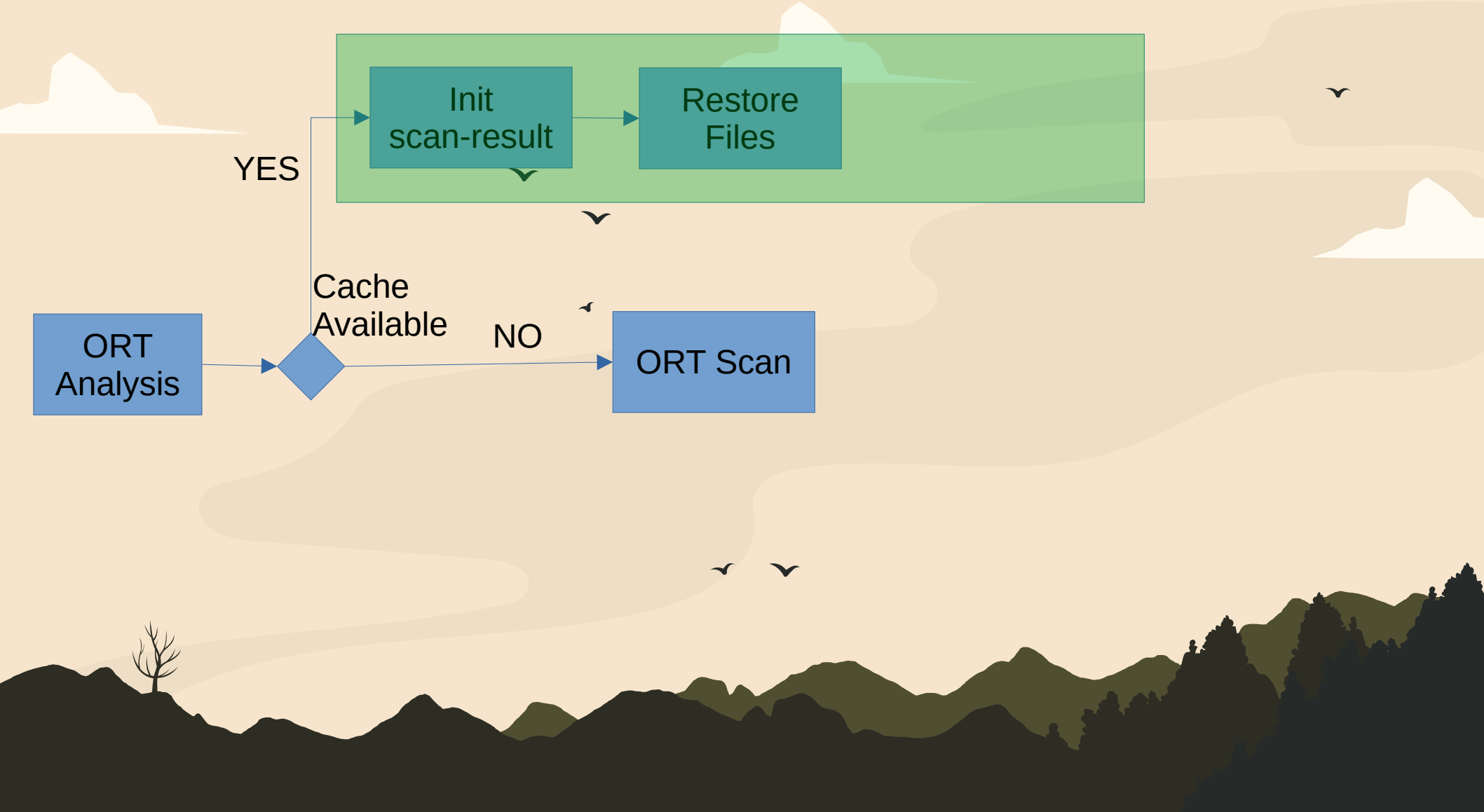

Init
scan-result
YES
Cache
Available
ORT
Analysis
NO
ORT
Init scan-result
- Extends the analysis-result.json
- Sets files with their sha1 hash

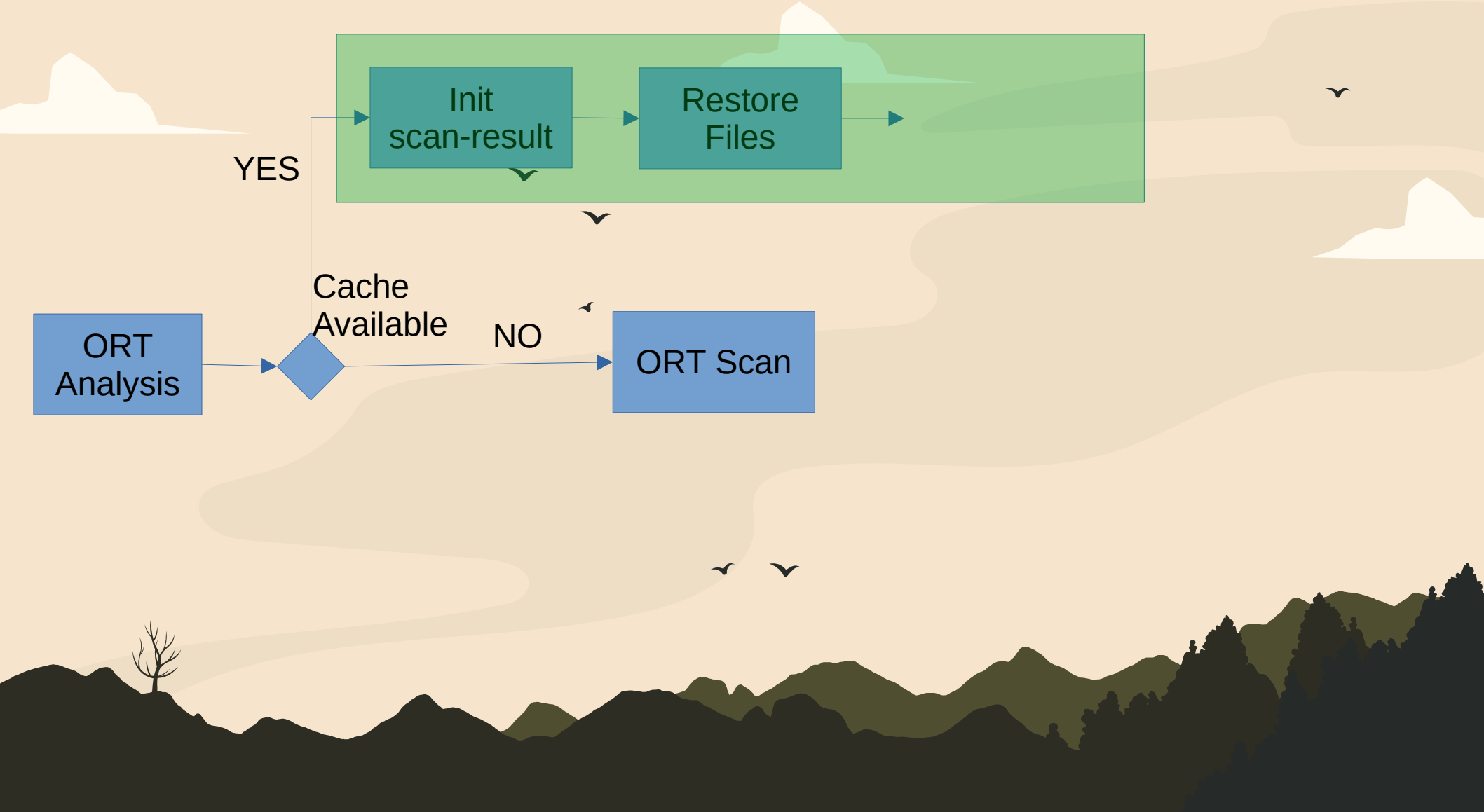

Init
scan-result
YES
Cache
Available
ORT
Analysis
NO
ORT Scan

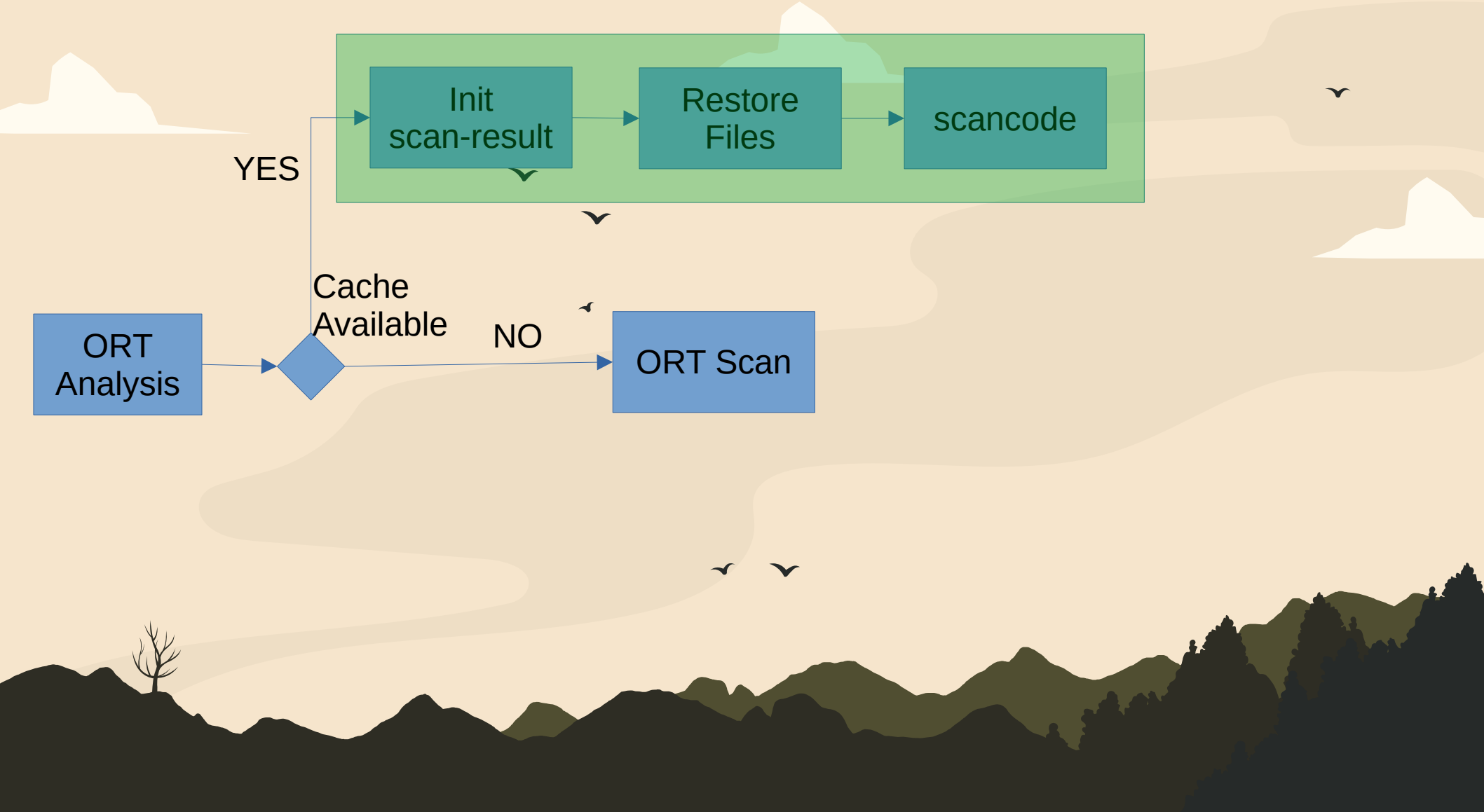

Init
scan-result
Restore
Files
YES
Cache
Available
ORT
Analysis
NO
ORT Scan

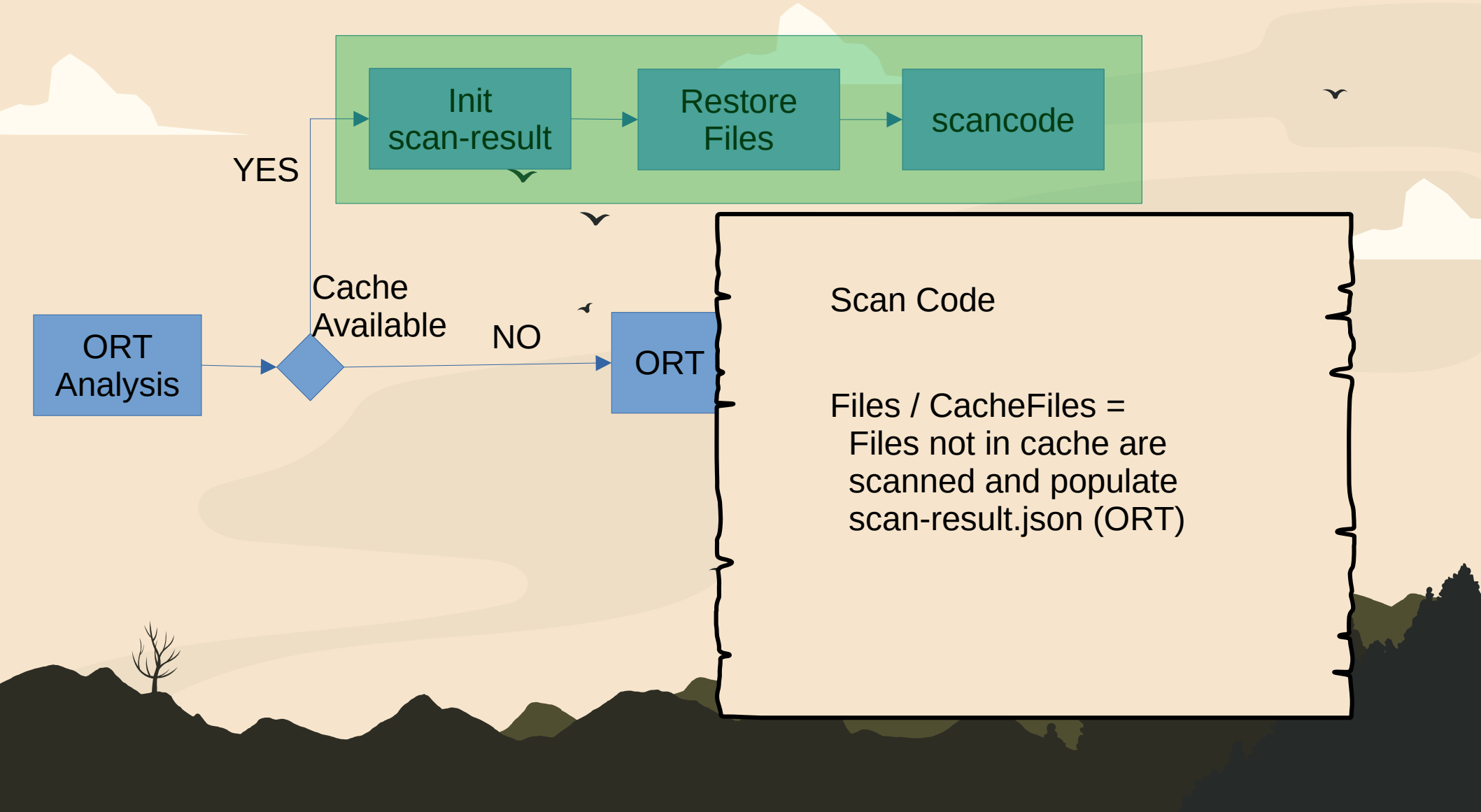

Init
scan-result
Restore
Files
YES
Cache
Available
ORT
Analysis
NO
ORT
Restore Files
- Fetch cached scan-result.json
- CacheFiles ∩Files =
Cached files must be
placed in
scan-result.json (ORT)

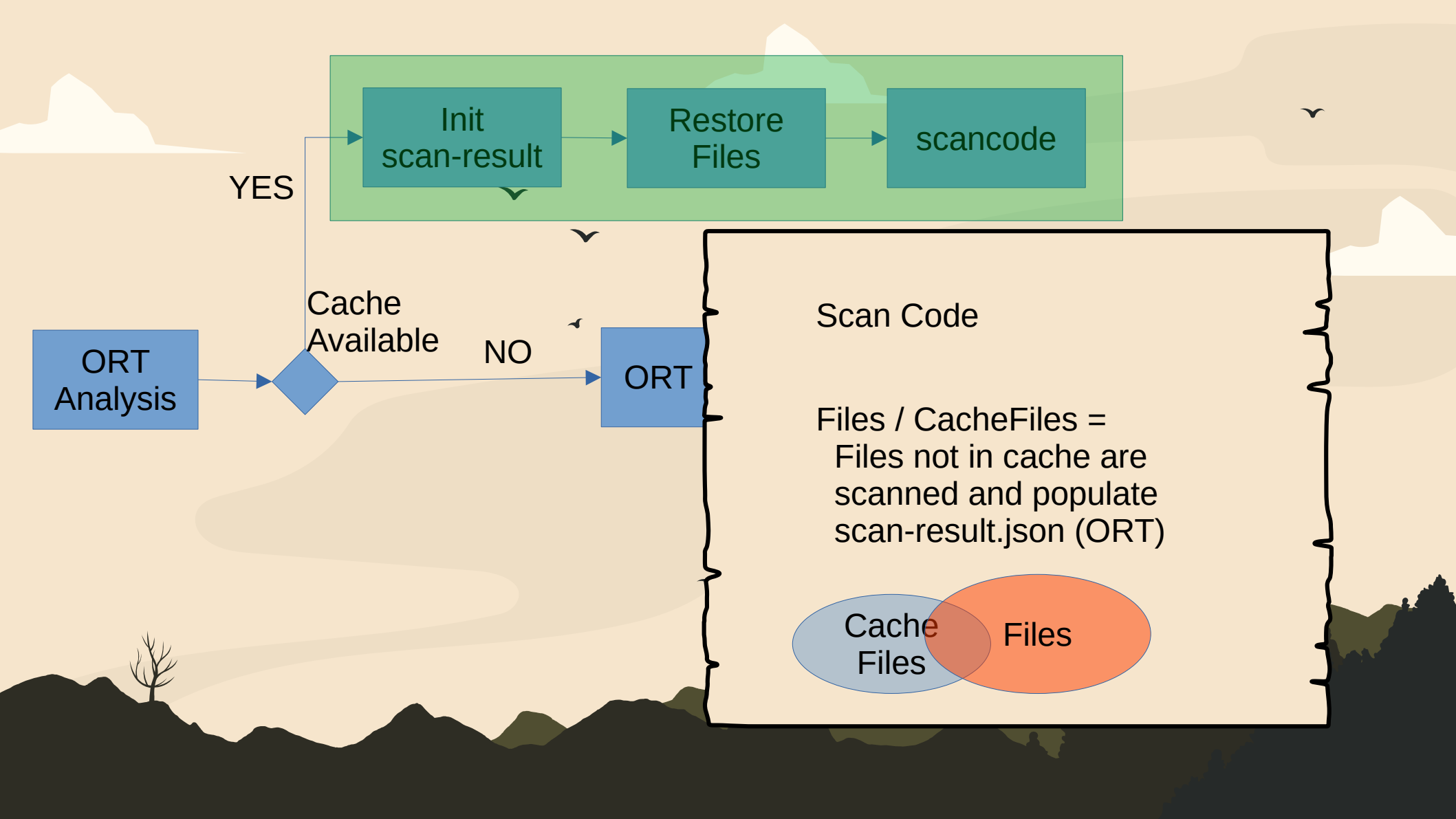

Init
scan-result
Restore
Files
YES
Cache
Available
ORT
Analysis
NO
ORT
Restore Files
- Fetch cached scan-result.json
- CacheFiles ∩Files =
Cached files must be
placed in
scan-result.json (ORT)
Cache
Files
Files

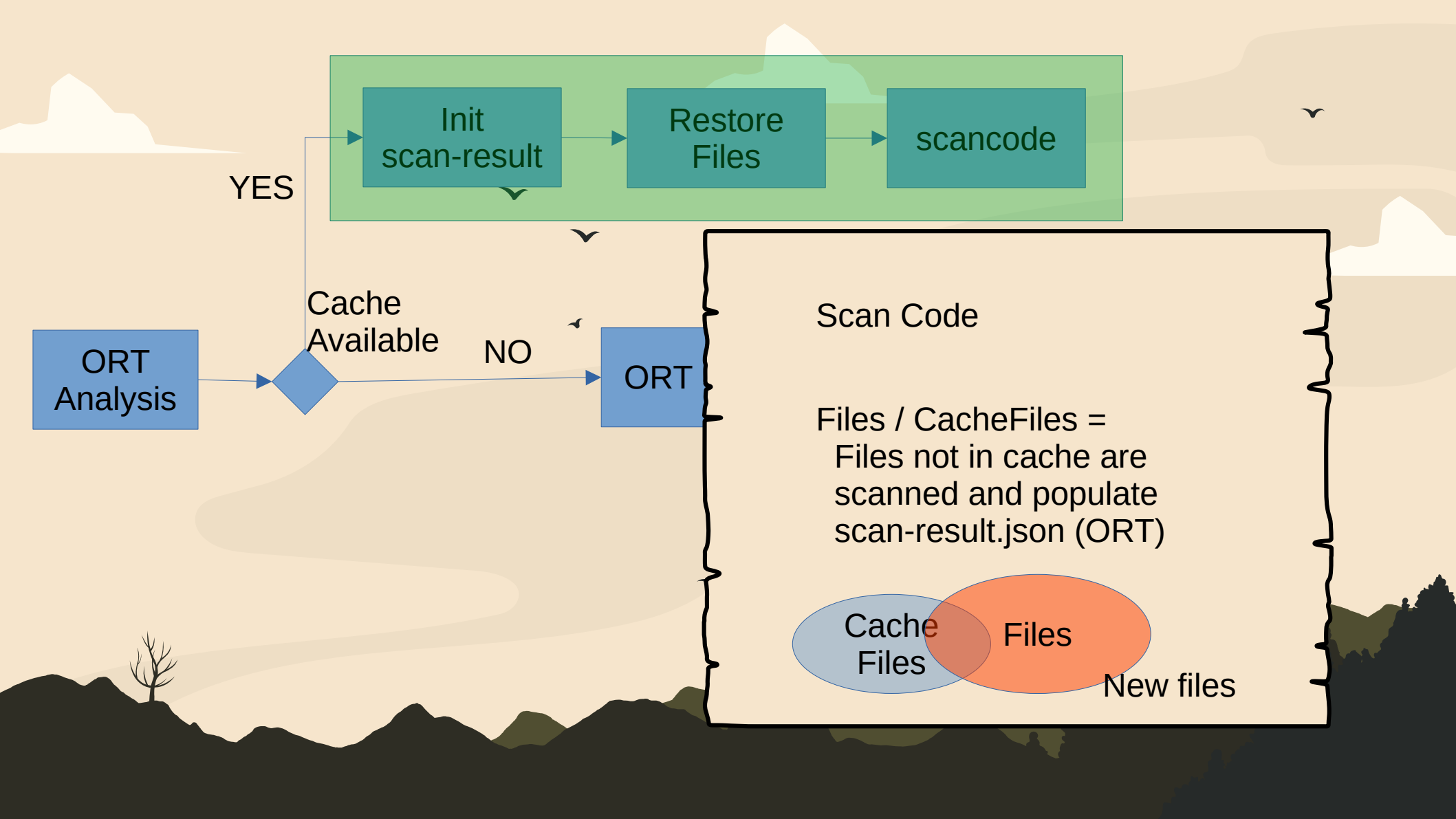

Init
scan-result
Restore
Files
YES
Cache
Available
ORT
Analysis
NO
ORT Scan

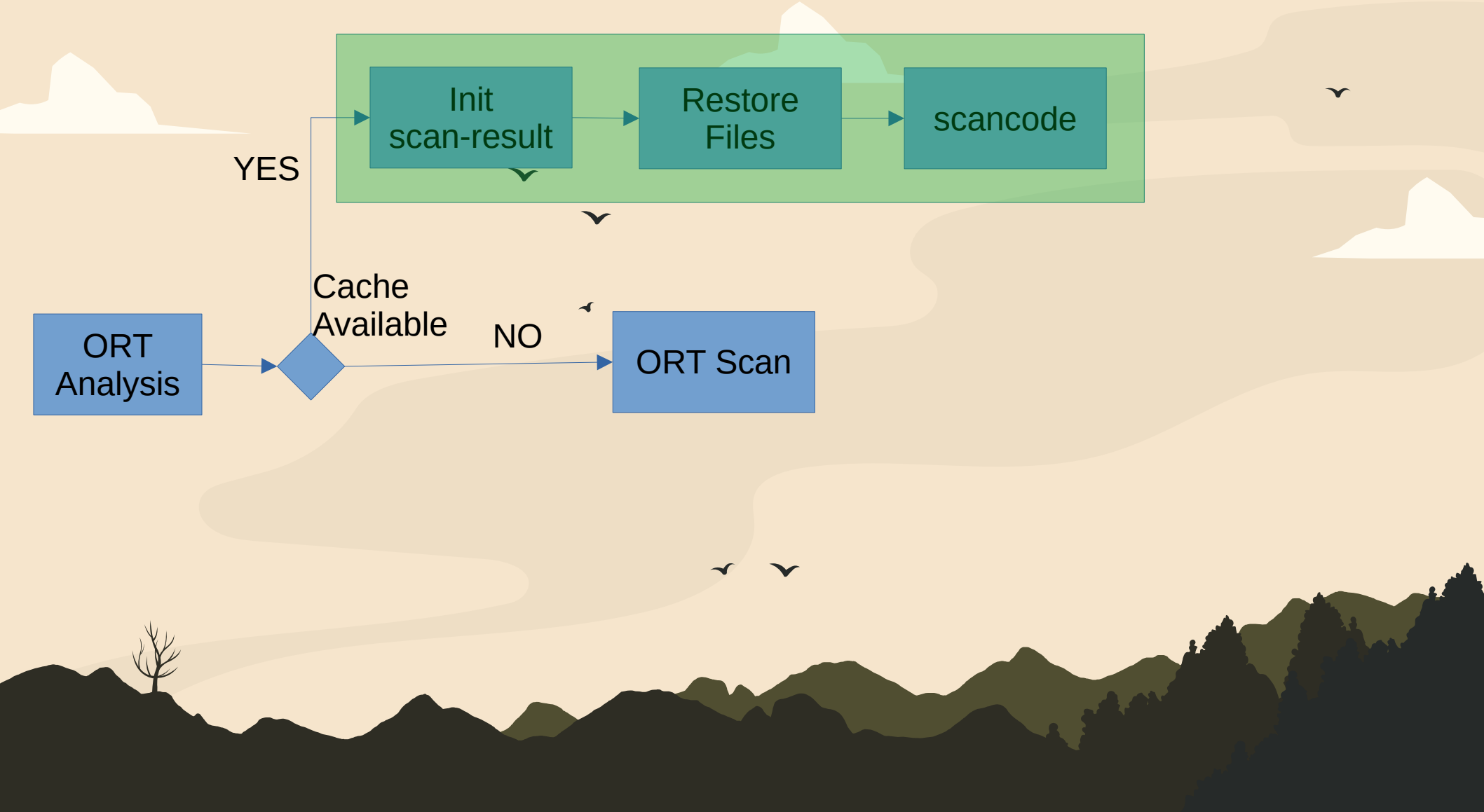

Init
scan-result
Restore
Files
YES
Cache
Available
ORT
Analysis
NO
ORT Scan

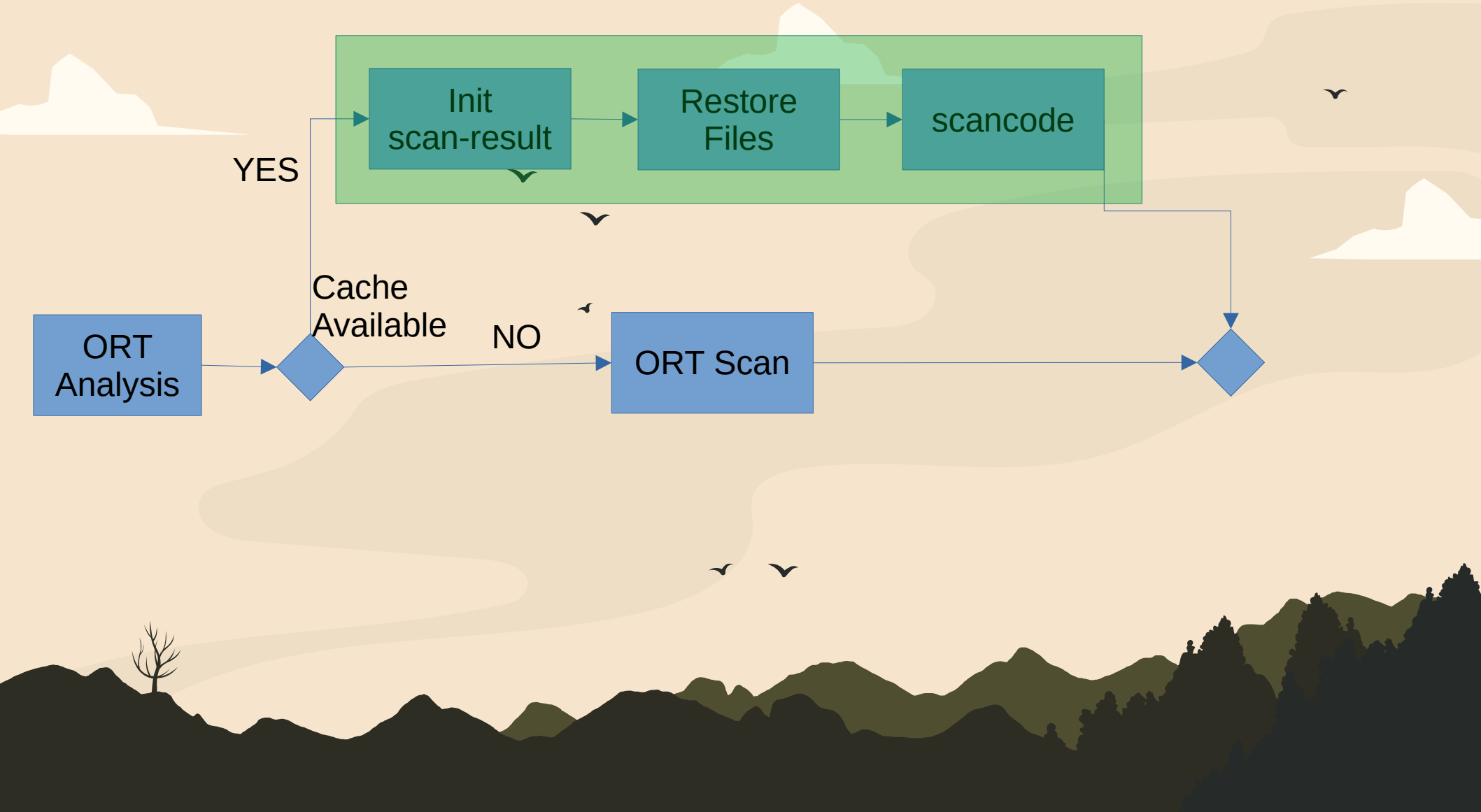

Init
scan-result
Restore
Files
scancode
YES
Cache
Available
ORT
Analysis
NO
ORT Scan

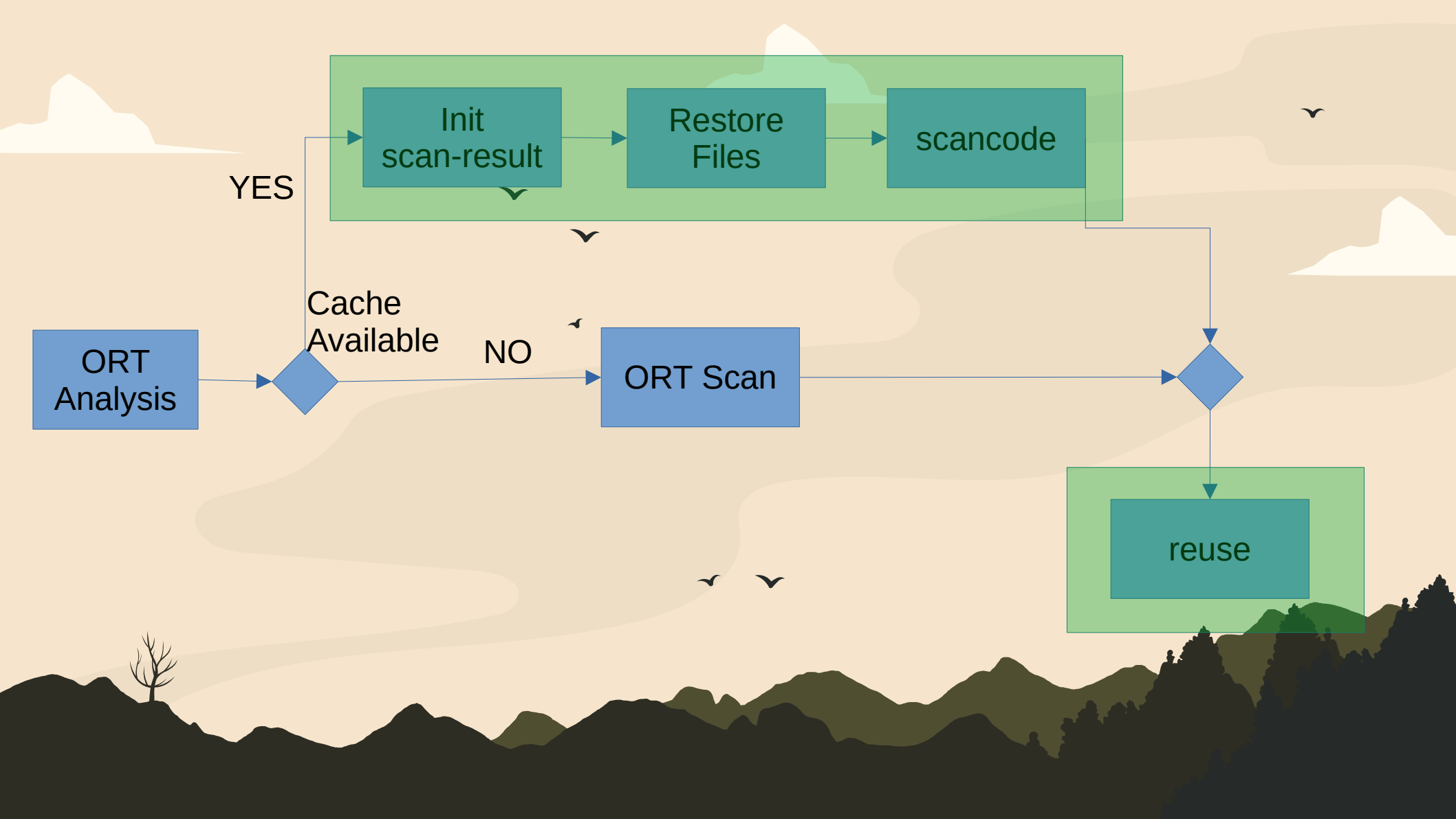

Init
scan-result
Restore
Files
scancode
YES
Cache
Available
NO
ORT
Analysis
ORT
Scan Code
Files / CacheFiles =
Files not in cache are
scanned and populate
scan-result.json (ORT)

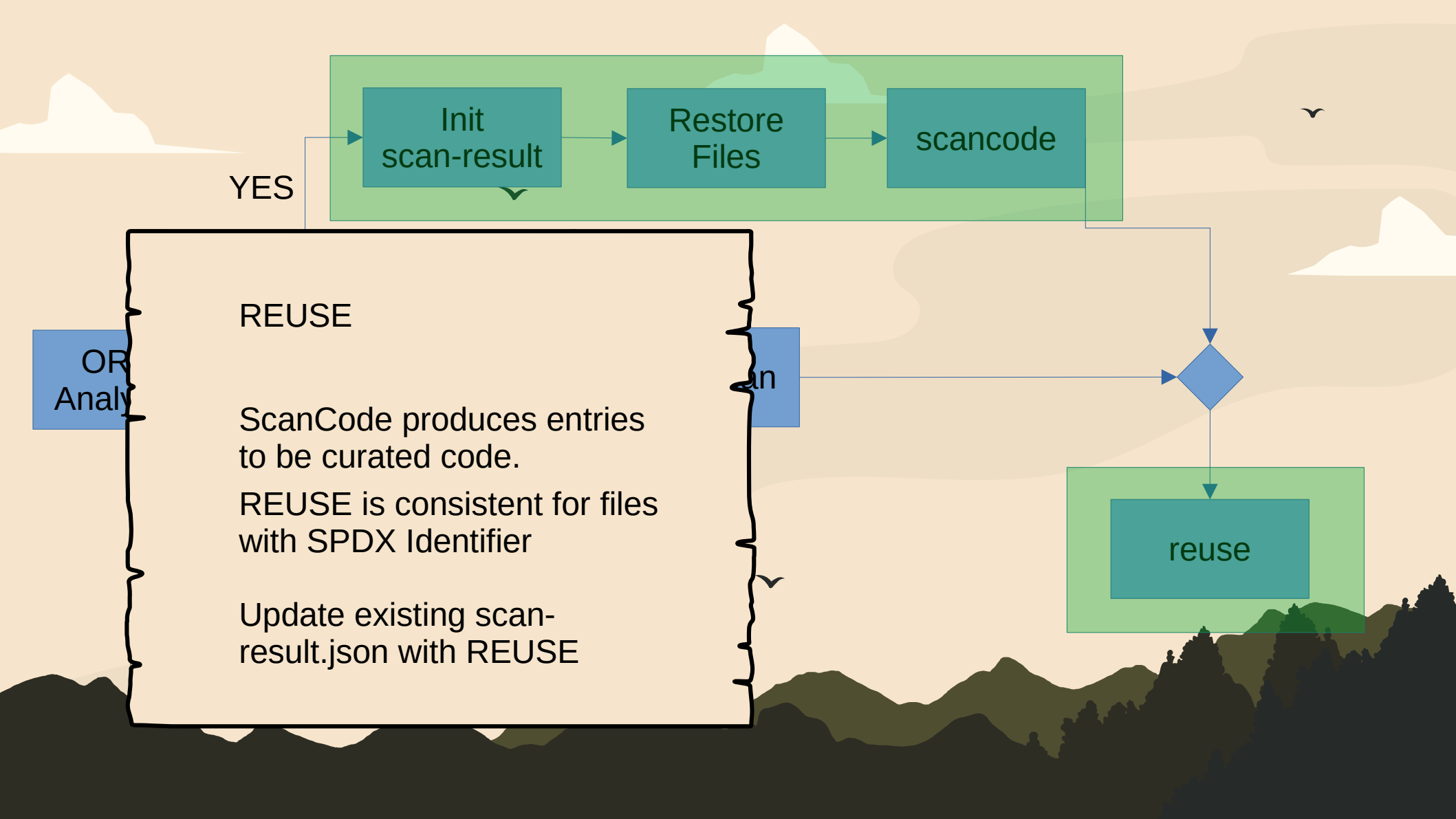

Init
scan-result
Restore
Files
scancode
YES
Cache
Available
ORT
Analysis
NO
ORT
Scan Code
Files / CacheFiles =
Files not in cache are
scanned and populate
scan-result.json (ORT)
Cache
Files
Files

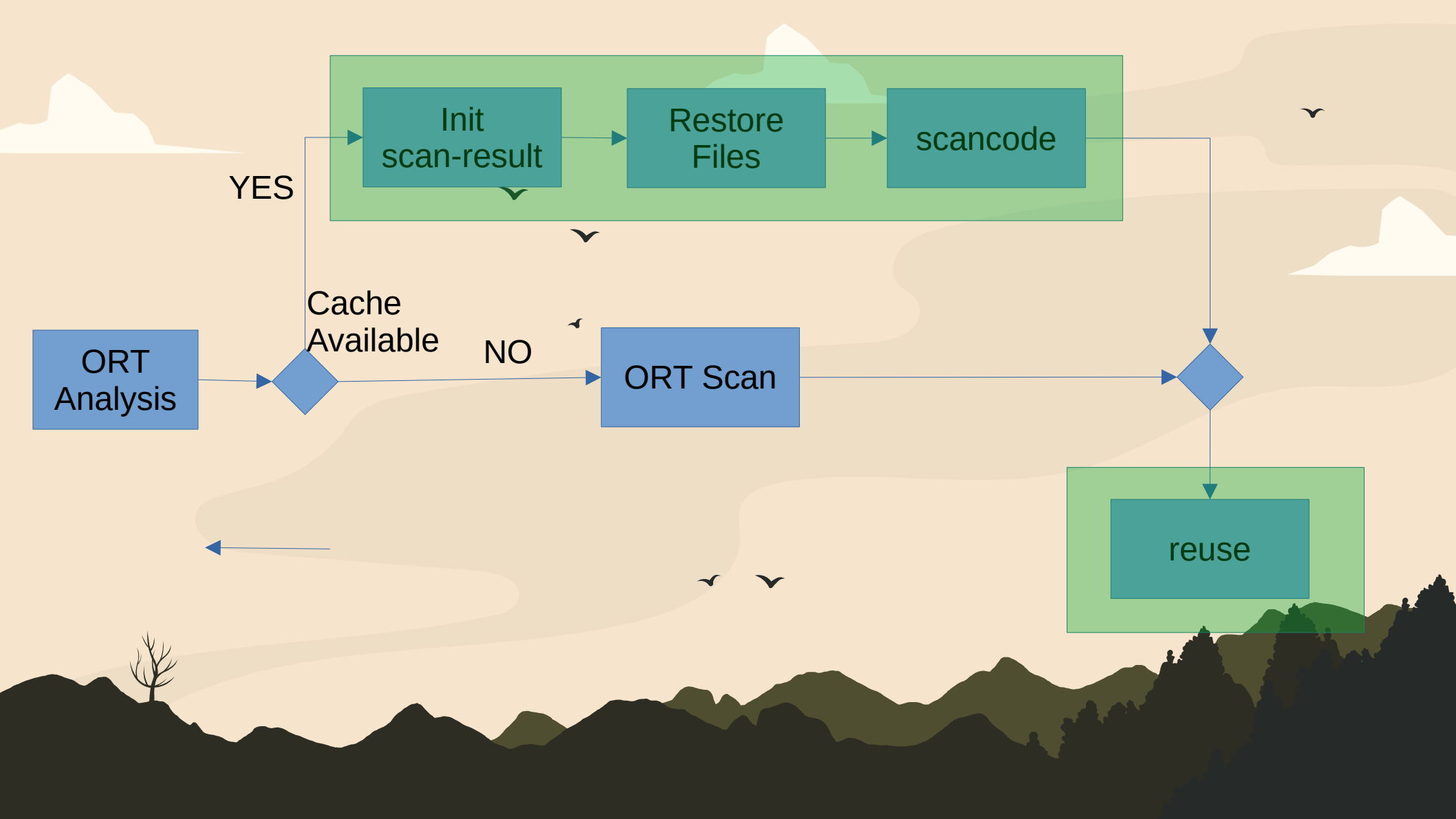

Init
scan-result
Restore
Files
scancode
YES
Cache
Available
ORT
Analysis
NO
ORT
Scan Code
Files / CacheFiles =
Files not in cache are
scanned and populate
scan-result.json (ORT)
Cache
Files
Files
New files

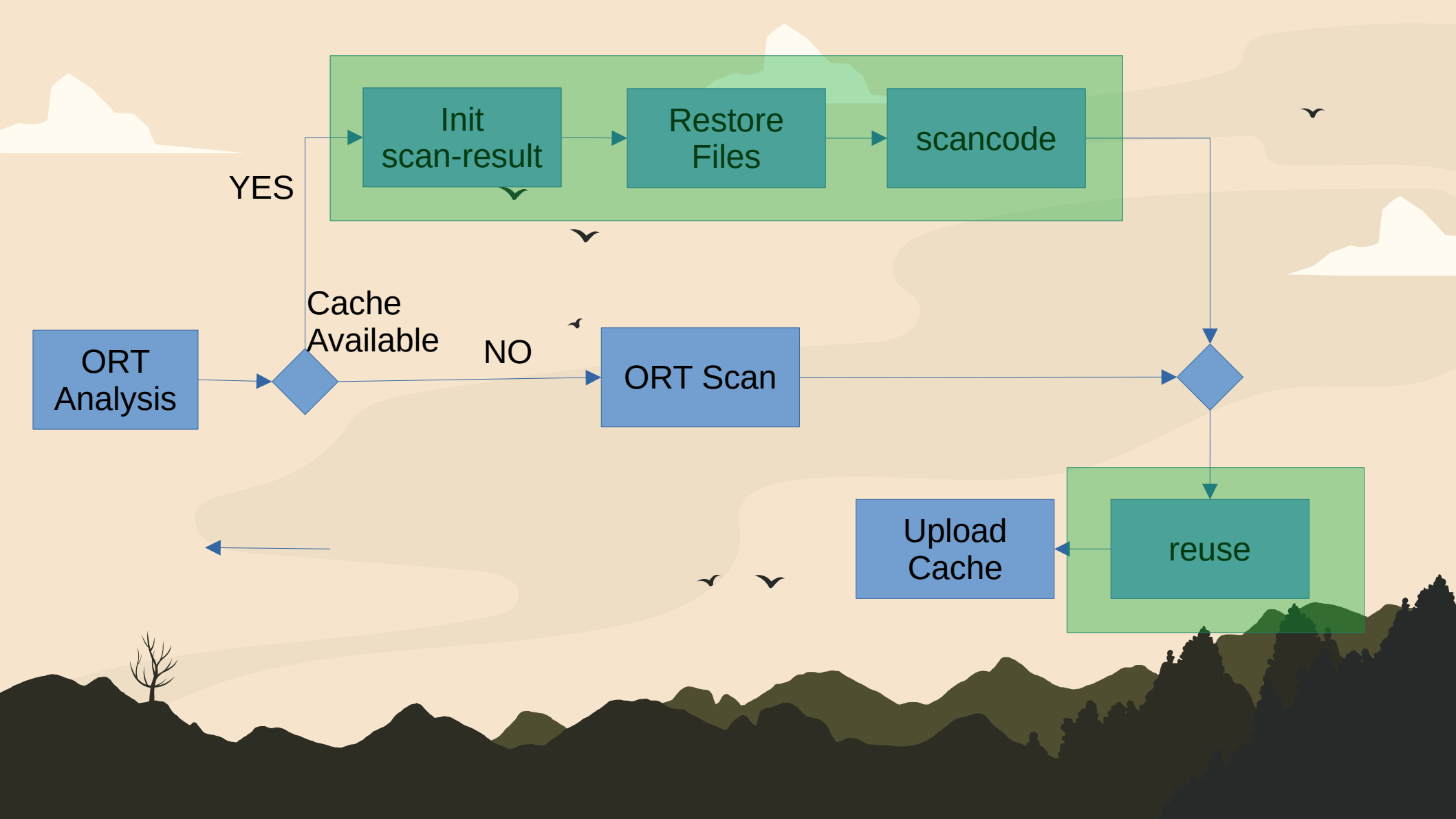

Init
scan-result
Restore
Files
scancode
YES
Cache
Available
ORT
Analysis
NO
ORT Scan

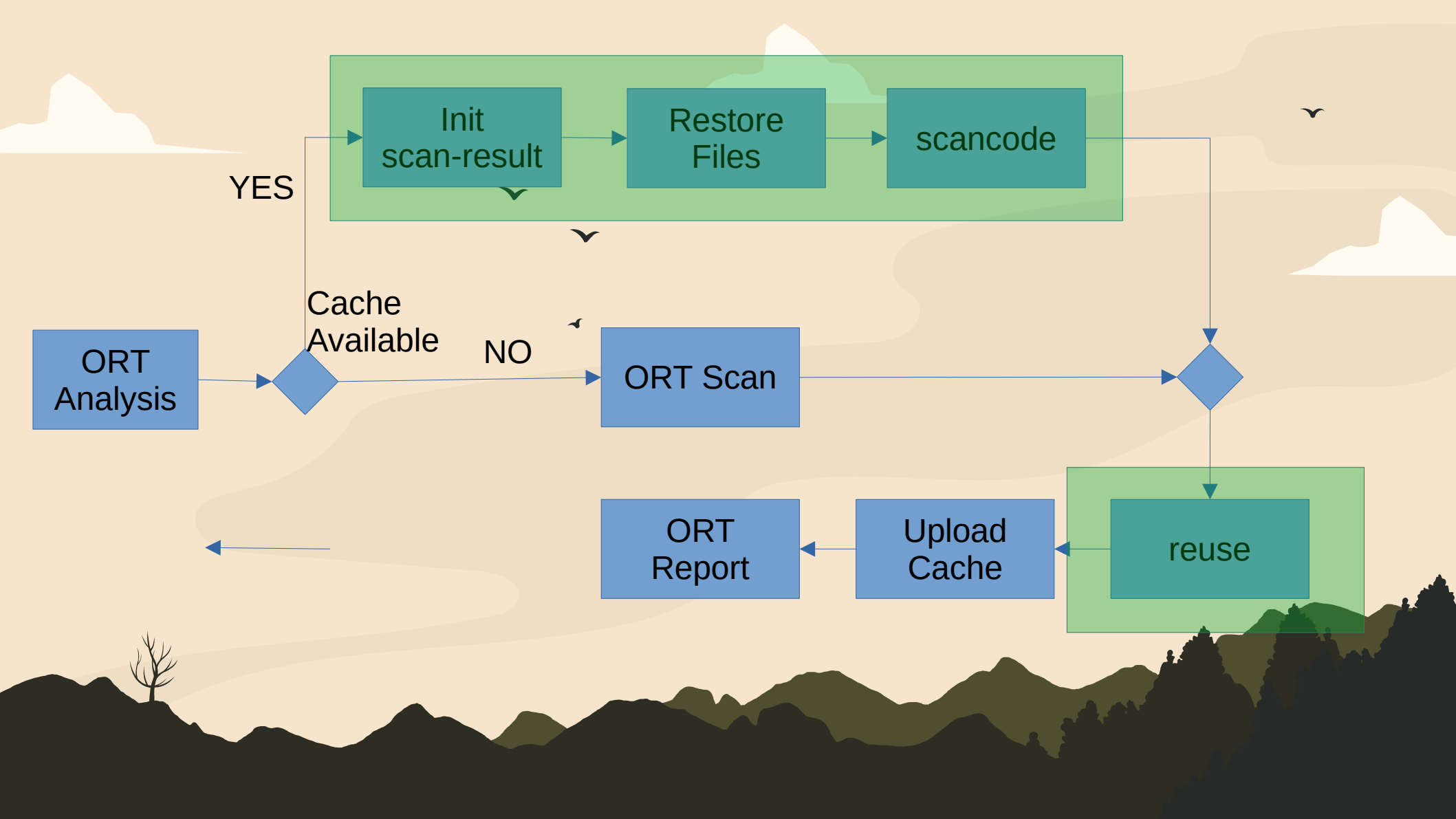

Init
scan-result
Restore
Files
scancode
YES
Cache
Available
NO
ORT
Analysis
ORT Scan

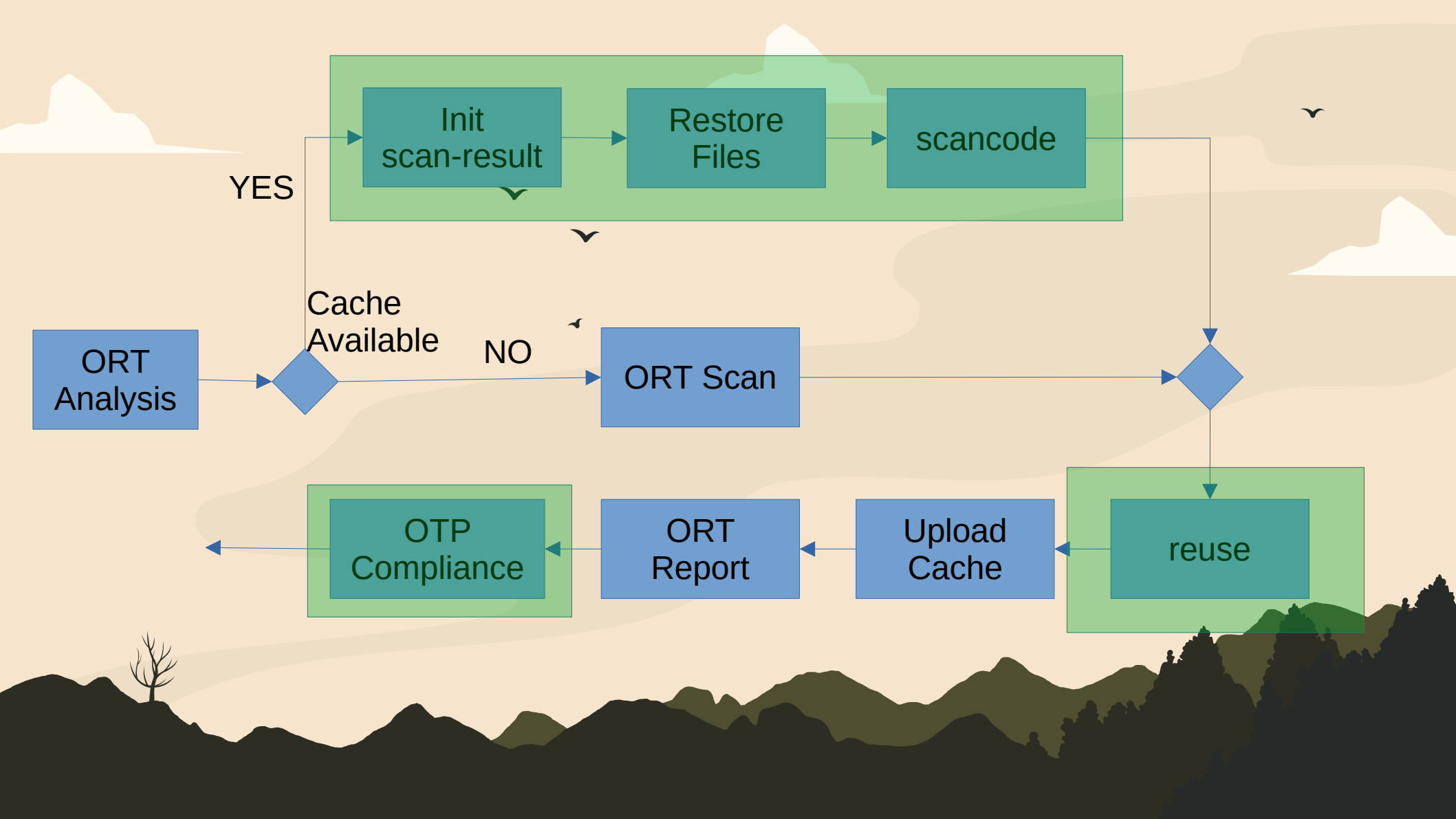

Init
scan-result
Restore
Files
scancode
YES
Cache
Available
ORT
Analysis
NO
ORT Scan
reuse


Init
scan-result
Restore
Files
scancode
YES
REUSE
ScanCode produces entries
to be curated code.
REUSE is consistent for files
with SPDX Identifier
Update existing scan-
result.json with REUSE
reuse


Init
scan-result
Restore
Files
scancode
YES
Cache
Available
ORT
Analysis
NO
ORT Scan
reuse


Init
scan-result
Restore
Files
scancode
YES
Cache
Available
ORT
Analysis
NO
ORT Scan
Upload
Cache
reuse


Init
scan-result
Restore
Files
scancode
YES
Cache
Available
NO
ORT
Analysis
ORT Scan
ORT
Report
Upload
Cache
reuse


Init
scan-result
Restore
Files
scancode
YES
Cache
Available
NO
ORT
Analysis
ORT Scan
OTP
Compliance
ORT
Report
Upload
Cache
reuse

Erlang Apps as Packages

Erlang Apps as Packages

```
175471    {
175472      "SPDXID": "SPDXRef-otp-ssh",
175473      "downloadLocation": "https://github.com/erlang/otp/release
175474      "externalRefs": [
175475        {
175476          "comment": "SSH-2 for Erlang/OTP",
175477          "referenceCategory": "PACKAGE-MANAGER",
175478          "referenceLocator": "pkg:otp/ssh@5.2.8",
175479          "referenceType": "purl"
175480        }
175481      ],
175482      "filesAnalyzed": true,
175483      "hasFiles": [
175484        "SPDXRef-File-8201",
175485        "SPDXRef-File-8202",
175486        "SPDXRef-File-8215",
175487        "SPDXRef-File-8216",
175488        "SPDXRef-File-8217",
175489        "SPDXRef-File-8218",
175490        "SPDXRef-File-8219",
175491        "SPDXRef-File-8220",
```

Erlang Apps as Packages

```
175471    {
175472      "SPDXID": "SPDXRef-otp-ssh",
175473      "downloadLocation": "https://github.com/erlang/otp/release
175474      "externalRefs": [
175475        {
175476          "comment": "SSH-2 for Erlang/OTP",
175477          "referenceCategory": "PACKAGE-MANAGER",
175478          "referenceLocator": "pkg:otp/ssh@5.2.8",
175479          "referenceType": "purl"
175480        }
175481      ],
175482      "filesAnalyzed": true,
175483      "hasFiles": [
175484        "SPDXRef-File-8201",
175485        "SPDXRef-File-8202",
175486        "SPDXRef-File-8215",
175487        "SPDXRef-File-8216",
175488        "SPDXRef-File-8217",
175489        "SPDXRef-File-8218",
175490        "SPDXRef-File-8219",
175491        "SPDXRef-File-8220",
```

Each package split by test, documentation, src

Each package split by test, documentation, src

```
175471      {
175472        "SPDXID": "SPDXRef-otp-ssh",
175473        "downloadLocation": "https://github.com/erlang/otp/release
175474        "externalRefs": [
175475          {
175476            "comment": "SSH-2 for Erlang/OTP",
175477            "referenceCategory": "PACKAGE-MANAGER",
175478            "referenceLocator": "pkg:otp/ssh@5.2.8",
175479            "referenceType": "purl"
175480          }
175481        ],
175482        "filesAnalyzed": true,
175483        "hasFiles": [
175484          "SPDXRef-File-8201",
175485          "SPDXRef-File-8202",
175486          "SPDXRef-File-8215",
175487          "SPDXRef-File-8216",
175488          "SPDXRef-File-8217",
175489          "SPDXRef-File-8218",
175490          "SPDXRef-File-8219",
175491          "SPDXRef-File-8220",
```

Each package split by test, documentation, src

```
188146    {
188147      "SPDXID": "SPDXRef-otp-ssh-documentation",
188148      "downloadLocation": "https://github.com/erlang/otp/releases",
188149      "externalRefs": [],
188150      "filesAnalyzed": true,
188151      "hasFiles": [
188152        "SPDXRef-File-8203",
188153        "SPDXRef-File-8204",
188154        "SPDXRef-File-8205",
188155        "SPDXRef-File-8206",
188156        "SPDXRef-File-8207",
188157        "SPDXRef-File-8208",
188158        "SPDXRef-File-8209",
188159        "SPDXRef-File-8210",
188160        "SPDXRef-File-8211",
188161        "SPDXRef-File-8212",
188162        "SPDXRef-File-8213",
188163        "SPDXRef-File-8214"
188164      ],
188165      "homepage": "https://www.erlang.org",
188166      "licenseConcluded": "Apache-2.0",
188167      "licenseDeclared": "Apache-2.0",
188168      "licenseInfoFromFiles": [
188169        "Apache-2.0",
188170        "NONE"
188171      ],
```

Each package split by test, documentation, src

```
188146      {
188147        "SPDXID": "SPDXRef-otp-ssh-documentation",
188148        "downloadLocation": "https://github.com/erlang/otp/releases",
    {
      "relatedSpdxElement": "SPDXRef-otp-ssh",
      "relationshipType": "TEST_OF",
      "spdxElementId": "SPDXRef-otp-ssh-test"
    },
    {
      "relatedSpdxElement": "SPDXRef-otp-ssh",
      "relationshipType": "DOCUMENTATION_OF",
      "spdxElementId": "SPDXRef-otp-ssh-documentation"
    },
    {
      "relatedSpdxElement": "SPDXRef-Project-OTP",
      "relationshipType": "PACKAGE_OF",
      "spdxElementId": "SPDXRef-otp-ssh"
    },
188168        "licenseInfoFromFiles": [
188169          "Apache-2.0",
188170          "NONE"
188171        ],
```

Each package split by test, documentation, src

```
188146      {
188147        "SPDXID": "SPDXRef-otp-ssh-documentation",
188148        "downloadLocation": "https://github.com/erlang/otp/releases",
    {
      "relatedSpdxElement": "SPDXRef-otp-ssh",
      "relationshipType": "TEST_OF",
      "spdxElementId": "SPDXRef-otp-ssh-test"
    },
    {
      "relatedSpdxElement": "SPDXRef-otp-ssh",
      "relationshipType": "DOCUMENTATION_OF",
      "spdxElementId": "SPDXRef-otp-ssh-documentation"
    },
    {
      "relatedSpdxElement": "SPDXRef-Project-OTP",
      "relationshipType": "PACKAGE_OF",
      "spdxElementId": "SPDXRef-otp-ssh"
    },
188168        "licenseInfoFromFiles": [
188169          "Apache-2.0",
188170          "NONE"
188171        ],
```

Each package split by test, documentation, src

```
188146      {
188147        "SPDXID": "SPDXRef-otp-ssh-documentation",
188148        "downloadLocation": "https://github.com/erlang/otp/releases",
    {
      "relatedSpdxElement": "SPDXRef-otp-ssh",
      "relationshipType": "TEST_OF",
      "spdxElementId": "SPDXRef-otp-ssh-test"
    },
    {
      "relatedSpdxElement": "SPDXRef-otp-ssh",
      "relationshipType": "DOCUMENTATION_OF",
      "spdxElementId": "SPDXRef-otp-ssh-documentation"
    },
    {
      "relatedSpdxElement": "SPDXRef-Project-OTP",
      "relationshipType": "PACKAGE_OF",
      "spdxElementId": "SPDXRef-otp-ssh"
    },
188168        "licenseInfoFromFiles": [
188169          "Apache-2.0",
188170          "NONE"
188171        ],
```

Each package split by test, documentation, src

```
188146      {
188147        "SPDXID": "SPDXRef-otp-ssh-documentation",
188148        "downloadLocation": "https://github.com/erlang/otp/releases",
    {
      "relatedSpdxElement": "SPDXRef-otp-ssh",
      "relationshipType": "TEST_OF",
      "spdxElementId": "SPDXRef-otp-ssh-test"
    },
    {
      "relatedSpdxElement": "SPDXRef-otp-ssh",
      "relationshipType": "DOCUMENTATION_OF",
      "spdxElementId": "SPDXRef-otp-ssh-documentation"
    },
    {
      "relatedSpdxElement": "SPDXRef-Project-OTP",
      "relationshipType": "PACKAGE_OF",
      "spdxElementId": "SPDXRef-otp-ssh"
    },
188168        "licenseInfoFromFiles": [
188169          "Apache-2.0",
188170          "NONE"
188171        ],
```

Dependencies between Erlang Apps


```
},
{
  "relatedSpdxElement": "SPDXRef-otp-erts",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-kernel",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-stdlib",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-crypto",
  "relationshipType": "DEPENDS_ON",
```

Dependencies between Erlang Apps

```
{
  "relatedSpdxElement": "SPDXRef-otp-erts",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-kernel",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-stdlib",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-crypto",
  "relationshipType": "DEPENDS_ON",
```

Dependencies between Erlang Apps


```
},
{
  "relatedSpdxElement": "SPDXRef-otp-erts",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-kernel",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-stdlib",
  "relationshipType": "DEPENDS_ON",
  "spdxElementId": "SPDXRef-otp-ssh"
},
{
  "relatedSpdxElement": "SPDXRef-otp-crypto",
  "relationshipType": "DEPENDS_ON",
```

Vendor libraries as packages


```
{
  "SPDXID": "SPDXRef-otp-erts-zlib",
  "comment": "vendor package",
  "copyrightText": "Copyright (C) 1995-2024 Jean-loup Gailly and Mark Adler",
  "description": "interface of the 'zlib' general purpose compression library",
  "downloadLocation": "https://zlib.net/",
  "externalRefs": [
    {
      "comment": "interface of the 'zlib' general purpose compression library",
      "referenceCategory": "PACKAGE-MANAGER",
      "referenceLocator": "pkg:generic/zlib@1.3.1",
      "referenceType": "purl"
    }
  ],
  "filesAnalyzed": true,
  "hasFiles": [
    "SPDXRef-File-1364",
    "SPDXRef-File-1365",
    "SPDXRef-File-1366",
    "SPDXRef-File-1367",
    "SPDXRef-File-1368",
    "SPDXRef-File-1369",
    "SPDXRef-File-1370",
    "SPDXRef-File-1371",
```

Vendor libraries as packages


```
{
  "SPDXID": "SPDXRef-otp-erts-zlib",
  "comment": "vendor package",
  "copyrightText": "Copyright (C) 1995-2024 Jean-loup Gailly and Mark Adler",
  "description": "interface of the 'zlib' general purpose compression library",
  "downloadLocation": "https://zlib.net/",
  "externalRefs": [
    {
      "comment": "interface of the 'zlib' general purpose compression library",
      "referenceCategory": "PACKAGE-MANAGER",
      "referenceLocator": "pkg:generic/zlib@1.3.1",
      "referenceType": "purl"
    }
  ],
  "filesAnalyzed": true,
  "hasFiles": [
    "SPDXRef-File-1364",
    "SPDXRef-File-1365",
    "SPDXRef-File-1366",
    "SPDXRef-File-1367",
    "SPDXRef-File-1368",
    "SPDXRef-File-1369",
    "SPDXRef-File-1370",
    "SPDXRef-File-1371",
```

OPENCHAIN

Get Certified Participate Resources FAQ Partner

The Erlang/OTP Project Announces an OpenChain ISO/IEC 5230 Conformant Program

By Shane Coughlan | 2025-02-01 | Featured, News

ERLANG

The Erlang Ecosystem Foundation has set goals for 2025 of raising the community infrastructure, processes and tooling profile to accommodate the latest industry standards for supply chain and cybersecurity. The Erlang/OTP

First Steps using ORT in Elixir

1,214 Files

elixir / NOTICE

josevalim Add JSON encoding and decoding (#14021)

Code | Blame | 38 lines (27 loc) · 1.35 KB

```
LEGAL NOTICE INFORMATION
------------------------

All the files in this distribution are copyright to the terms below.

== lib/elixir/src/elixir_json.erl
== lib/elixir/src/elixir_parser.erl (generated by build scripts)

Copyright Ericsson AB 1996-2024

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

== All other files

Copyright 2012 Plataformatec
Copyright 2021 The Elixir Team

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   https://www.apache.org/licenses/LICENSE-2.0

```

1,214 Files

```yaml
- name: Run OSS Review Toolkit
  id: ort
  uses: oss-review-toolkit/ort-ci-github-action@1805edcf1f4f55f35ae6e4d2d9795ccfb29b6021
  with:
    image: ghcr.io/oss-review-toolkit/ort-minimal:54.0.0
    run: >-
      labels,
      cache-dependencies,
      cache-scan-results,
      analyzer,
      scanner,
      advisor,
      evaluator,
      reporter,
      ${{ inputs.upload-reports == 'true' && 'upload-results' || '' }}
    fail-on: "${{ inputs.fail-on-violation == 'true' && 'violations,issues' || '' }}"
    report-formats: "${{ inputs.report-formats }}"
    ort-cli-report-args: >-
      -O CycloneDX=output.file.formats=json,xml
      -O SpdxDocument=outputFileFormats=JSON,YAML
    sw-version: "${{ inputs.version }}"
```

1,214 Files

```
val whitelistedLicenses = listOf(
  // License for Elixir & Imported Erlang Projects
  "Apache-2.0",
  // License for the Elixir Logo
  "LicenseRef-elixir-trademark-policy",
  "LicenseRef-scancode-elixir-trademark-policy",
  // License for included Unicode Files
  "LicenseRef-scancode-unicode",
  // DCO for committers
  "LicenseRef-scancode-dco-1.1"
).map { SpdxSingleLicenseExpression.parse(it) }.toSet()

fun PackageRule.howToFixDefault() = """
      * Check if this license violation is intended
      * Adjust evaluation rules in `.ort/config/evaluator.rules.kts`
    """.trimIndent()

fun PackageRule.LicenseRule.isHandled() =
  object : RuleMatcher {
    override val description = "isHandled($license)"

    override fun matches() = license in whitelistedLicenses
```

1,214 F

```
SPDXID: "SPDXRef-DOCUMENT"
spdxVersion: "SPDX-2.2"
creationInfo:
  created: "2025-02-05T12:29:35Z"
  creators:
  - "Organization: The Elixir Team"
  licenseListVersion: "3.9"
name: "elixir"
dataLicense: "CC0-1.0"
documentNamespace: "https://github.com/elixir-lang/elixir"
documentDescribes:
- "SPDXRef-Project-elixir-lang"
packages:
- SPDXID: "SPDXRef-Project-elixir-lang"
  summary: "About Elixir is a dynamic, functional language for building scalable and maintainable applications"
  copyrightText: "Copyright (c) 2012 Plataformatec. Copyright (c) 2021 The Elixir Team. All Rights Reserved."
  downloadLocation: "git+https://github.com/elixir-lang/elixir.git"
  filesAnalyzed: false
  homepage: "https://elixir-lang.org/"
  licenseConcluded: "Apache-2.0 AND LicenseRef-scancode-unicode AND LicenseRef-elixir-trademark-policy"
  licenseDeclared: "Apache-2.0 AND LicenseRef-scancode-unicode AND LicenseRef-elixir-trademark-policy"
  name: "elixir-lang"
  originator: "Organization: The Elixir Team"
  supplier: "Organization: The Elixir Team"
  packageFileName: "."
  externalRefs:
    - referenceCategory: PACKAGE-MANAGER
      referenceType: "purl"
      referenceLocator: "pkg:github/elixir-lang/elixir"
      comment: "GitHub PURL"
  # elixir-version-insert
- SPDXID: "SPDXRef-Package-eex"
  summary: "About Elixir is a dynamic, functional language for building scalable and maintainable applications"
  copyrightText: "Copyright (c) 2012 Plataformatec. Copyright (c) 2021 The Elixir Team. All Rights Reserved."
  downloadLocation: "git+https://github.com/elixir-lang/elixir.git#lib/eex"
  filesAnalyzed: false
  homepage: "https://elixir-lang.org/"
  licenseConcluded: "Apache-2.0"
  licenseDeclared: "Apache-2.0"
  name: "eex"
  originator: "Organization: The Elixir Team"
  supplier: "Organization: The Elixir Team"
  packageFileName: "./lib/eex"
```

Announcing Elixir OpenChain ISO/IEC 5230 Certification

By Shane Coughlan | 2025-02-26 | Featured, News

The Elixir Project is pleased to share that the Elixir project now complies with OpenChain ISO/IEC 5230, the international standard for open source license compliance. This step aligns with broader

ĖEF + ORT Future

- EEF would like to recommend ORT to our community

- Looking forward to explore SBOM as first class input

- Integration of package managers with ORT is not closed

Community effort

Questions?

Kiko Fernandez-Reyes, PhD

Member of Erlang/OTP

Board member of
Erlang Ecosystem Foundation