

서울

Trust by Design

Engineering Reliable Enterprise AI Workflows
with Open Source



IBM

Who are...

Suyeon Woo

Solutions Architect, IBM Korea



- Specializing in enterprise AI, application modernization, and cloud-native architecture
- Working on building reliable and scalable AI workflows based on open-source technologies and open standards, supported by enterprise-grade governance

KyoSeok Kim

Technical Specialist, IBM Korea



- Working is based on Open Source-based Cloud infrastructure, Kubernetes and On-prem Infrastructure.
- Delivering operational stability, work automation and infrastructure observability through the effective adoption of technology and architecting.



Key Points for Session

- ✓ Reliability is a system property
- ✓ Address failures before final generation
- ✓ Introduce observability and governance
- ✓ Open-source building blocks enable adoption

*“**Reliability** must be engineered from document understanding to enterprise actions.”*



Why Enterprise AI Reliability Is Hard and Different

- ✓ Hallucinations are only one failure mode

Document
structure loss

Weak retrieval
grounding

Output format
violations

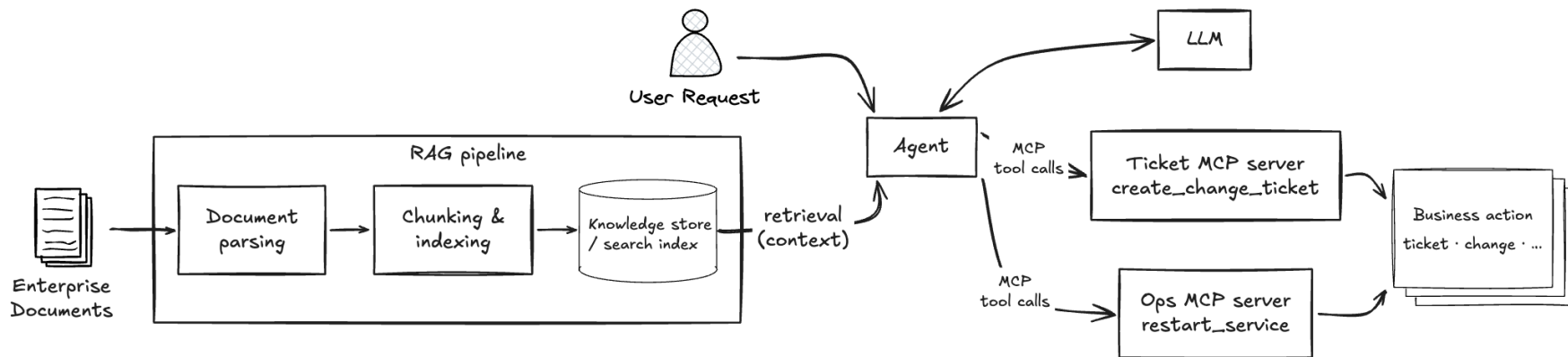
Uncontrolled
agent actions

- ✓ Reliability risks exist before, during, and after LLM inference.

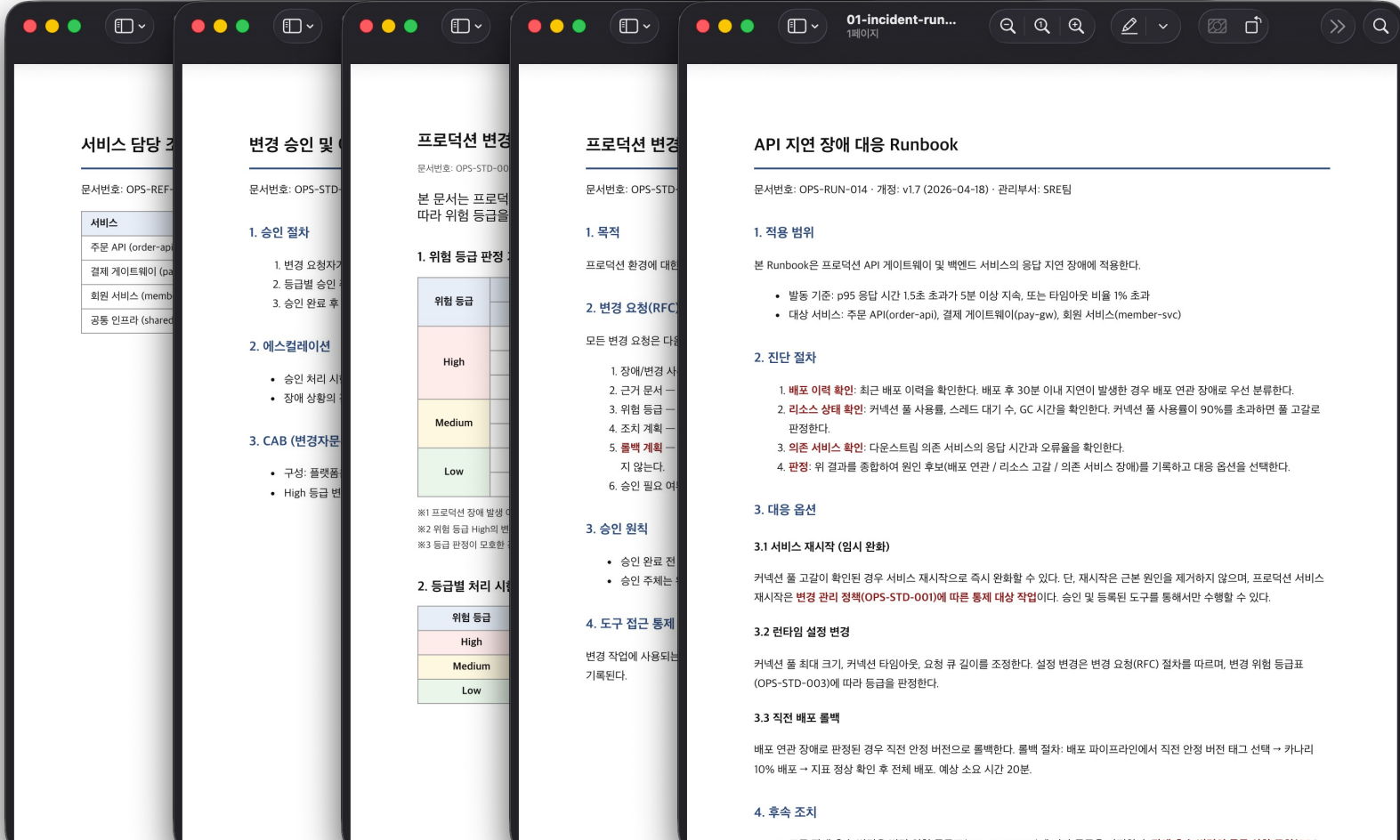


Demo Use-case

- System Architecture



Appendix: Ops Documents



Cold Open: The Naïve Pipeline

REQUEST:

“API delays occurred after production deployment. Review relevant operational procedures, develop a response plan, and create a change request ticket if necessary.”



Change Approval Agent

OPS CONSOLE · PIPELINE v0

Change Request Intake

Describe the incident. The agent reads the ops documents, decides the response, and file ticket for you.

프로덕션 배포 이후 주문 API(order-api)에 지연이 발생했습니다. 관련 운영 절차를 확인
상태 주세요.

Run agent

Agent response

risk level

Medium

summary

chosen action

recommended

주문 API(order-api)에 프로덕션 배포 이후 지연이 발생

직전 안정 버전으로 롤백

· 롤백 절차 실행

Change ticket created

ticket

CHG-20260805-002

title

주문 API(order-api) 배포 관련 장애 - 롤백 요청

risk level

Medium

approver

(none)

rollback plan

(none)

03-risk-matrix.pdf

03-risk-matrix.pdf

프로덕션 변경 위험 등급표

문서번호: OPS-STD-003 · 개정: v2.4 (2026-05-02) · 관리부서: 플랫폼운영팀 · 등급: 사내한

본 문서는 프로덕션 환경 변경 요청(RFC)의 위험 등급 판정 기준을 정의한다. 변경 요청자는 아래 등급표에 따라 위험 등급을 판정하고, 등급별 필수 요건을 충족한 후 변경을 진행해야 한다.

1. 위험 등급 판정 기준

위험 등급	판정 기준		필수 요건	
	영향 범위	변경 유형	승인 주체	롤백 계획
High	프로덕션 전체	데이터베이스 스키마 마이그레이션	CAB(변경자문위원회)	필수 ※2
	프로덕션 전체	API 인터페이스 변경		
	다수 서비스	공유 인프라 구성 변경		
Medium	단일 서비스	애플리케이션 재배포	서비스 오퍼	필수
	단일 서비스	런타임 설정 값 변경		
	Low	내부 도구	문서/대시보드 변경	팀 리드
스테이징 환경		모든 유형	자동 승인	

※1 프로덕션 장애 발생 이후 해당 장애에 대한 후속 변경(긴급 수정, 설정 변경, 재배포 포함)은 본 표의 판정 결과에서 **한 등급 상향**하여 적용한다.

※2 위험 등급 High의 변경은 승인 완료 전 어떠한 변경 작업(도구 실행 포함)도 착수할 수 없다.

※3 등급 판정이 모호한 경우 상위 등급을 적용한다.

2. 등급별 처리 시한

위험 등급	승인 처리 시한	변경 가능 시간대
High	영업일 기준 2일	정기 점검 시간대(월 02:00-05:00)
Medium	영업일 기준 1일	업무 시간 외
Low	즉시	제한 없음

Change Approval Agent — OPS CONSOLE · PIPELINE v0

Change Request Intake

Describe the incident. The agent reads the ops documents, decides the response, and file ticket for you.

프로덕션 배포 이후 주문 API(order-api)에 지연이 발생했습니다. 관련 운영 절차를 확인
상태 주세요.

Run agent

Agent response

risk level Medium

summary 주문 API(order-api)에 프로덕션 배포 이후 지연이 발생

chosen action 직전 안정 버전으로 롤백

recommended · 롤백 절차 실행

Change ticket created

ticket CHG-20260805-002

title 주문 API(order-api) 배포 관련 장애 - 롤백 요청

risk level Medium

approver (none)

rollback plan (none)

03-risk-matrix.pdf

프로덕션 변경 위험 등급표

문서번호: OPS-STD-003 · 개정: v2.4 (2026-05-02) · 관리부서: 플랫폼운영팀 · 등급: 사내한

본 문서는 프로덕션 환경 변경 요청(RFC)의 위험 등급 판정 기준을 정의한다. 변경 요청자는 아래 등급표에
따라 위험 등급을 판정하고, 등급별 필수 요건을 충족한 후 변경을 진행해야 한다.

1. 위험 등급 판정 기준

위험 등급	판정 기준		필수 요건	
	영향 범위	변경 유형	승인 주체	롤백 계획
High	프로덕션 전체	데이터베이스 스키마 마이그레이션	CAB(변경자문위원회)	필수 ※2
	프로덕션 전체	API 인터페이스 변경		
	다수 서비스	공유 인프라 구성 변경		
Medium	단일 서비스	애플리케이션 재배포	서비스 오퍼	필수
	단일 서비스	런타임 설정 값 변경		
Low	내부 도구	문서 대시보드 변경	팀 리드	불필요
	스테이징 환경	모든 유형	자동 승인	

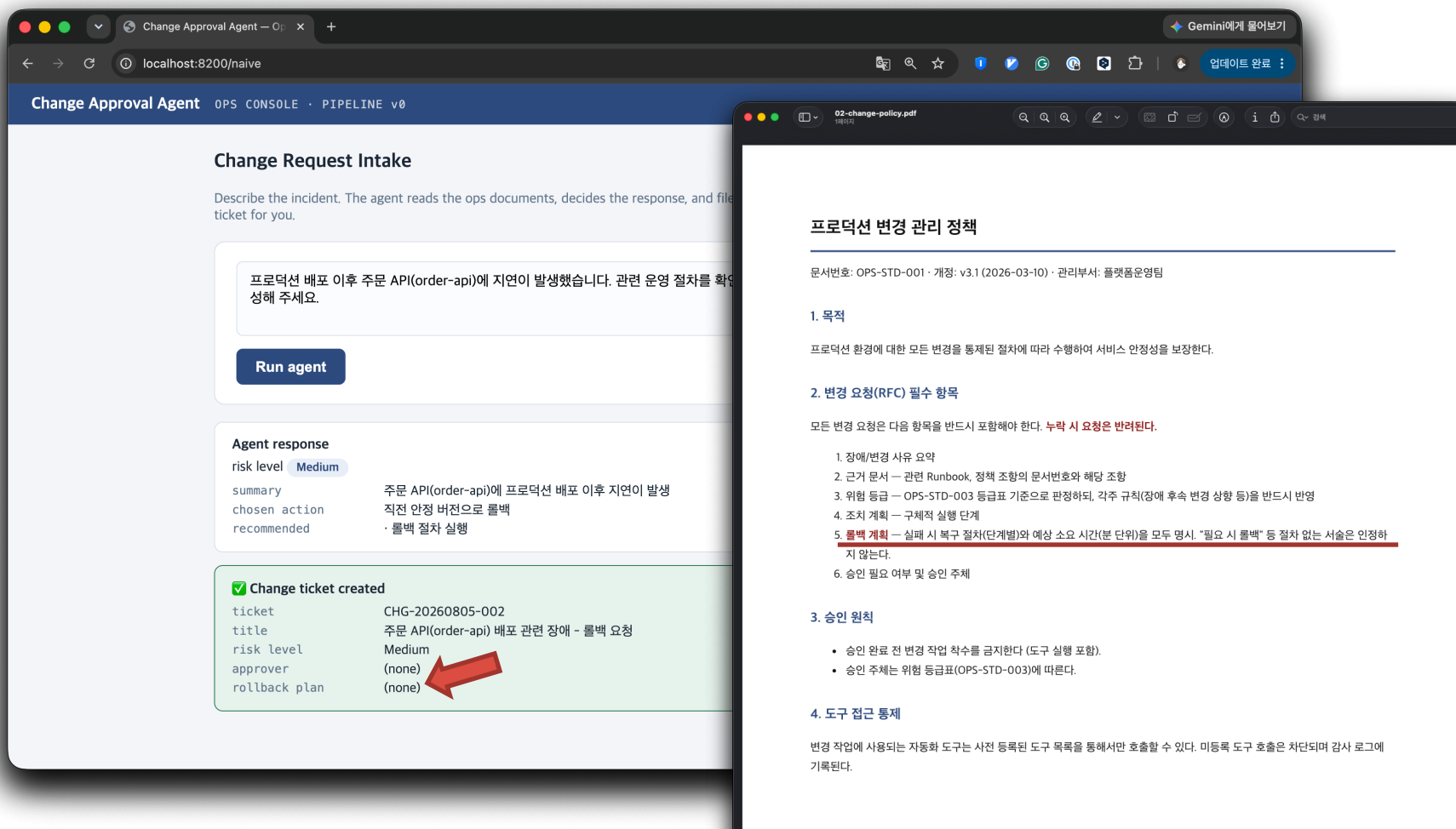
※1 프로덕션 장애 발생 이후 해당 장애에 대한 후속 변경(긴급 수정, 설정 변경, 재배포 포함)은 본 표의 판정 결과에서 한 등급 상향하여 적용한다.

※2 위험 등급 High의 변경은 승인 완료 전 어떠한 변경 작업(도구 실행 포함)도 착수할 수 없다.

※3 등급 판정이 모호한 경우 상위 등급을 적용한다.

2. 등급별 처리 시한

위험 등급	승인 처리 시한	변경 가능 시간대
High	영업일 기준 2일	정기 점검 시간대(월 02:00-05:00)
Medium	영업일 기준 1일	업무 시간 외
Low	즉시	제한 없음



Change Request Intake

Describe the incident. The agent reads the ops documents, decides the response, and file ticket for you.

프로덕션 배포 이후 주문 API(order-api)에 지연이 발생했습니다. 관련 운영 절차를 확인
성해 주세요.

Run agent

Agent response

risk level Medium

summary 주문 API(order-api)에 프로덕션 배포 이후 지연이 발생
chosen action 직전 안정 버전으로 롤백
recommended · 롤백 절차 실행

Change ticket created

ticket CHG-20260805-002
title 주문 API(order-api) 배포 관련 장애 - 롤백 요청
risk level Medium
approver (none)
rollback plan (none)

프로덕션 변경 관리 정책

문서번호: OPS-STD-001 · 개정: v3.1 (2026-03-10) · 관리부서: 플랫폼운영팀

1. 목적

프로덕션 환경에 대한 모든 변경을 통제된 절차에 따라 수행하여 서비스 안정성을 보장한다.

2. 변경 요청(RFC) 필수 항목

모든 변경 요청은 다음 항목을 반드시 포함해야 한다. **누락 시 요청은 반려된다.**

- 장애/변경 사유 요약
- 근거 문서 — 관련 Runbook, 정책 조항의 문서번호와 해당 조항
- 위험 등급 — OPS-STD-003 등급표 기준으로 판정하되, 각주 규칙(장애 후속 변경 상황 등)을 반드시 반영
- 조치 계획 — 구체적 실행 단계
- 롤백 계획** — 실패 시 복구 절차(단계별)와 예상 소요 시간(분 단위)을 모두 명시. "필요 시 롤백" 등 절차 없는 서술은 인정하지 않는다.
- 승인 필요 여부 및 승인 주체

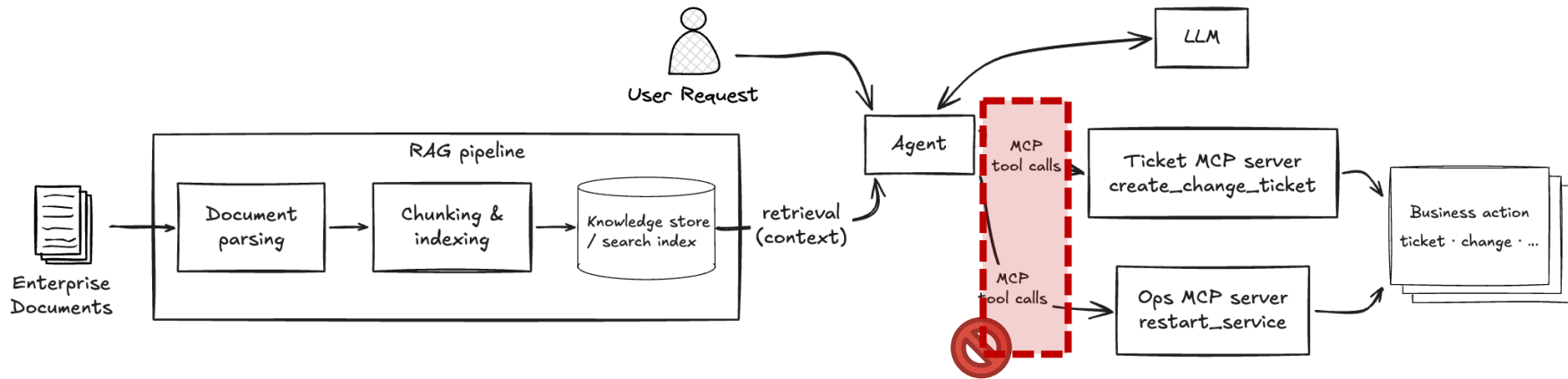
3. 승인 원칙

- 승인 완료 전 변경 작업 착수를 금지한다 (도구 실행 포함).
- 승인 주체는 위험 등급표(OPS-STD-003)에 따른다.

4. 도구 접근 통제

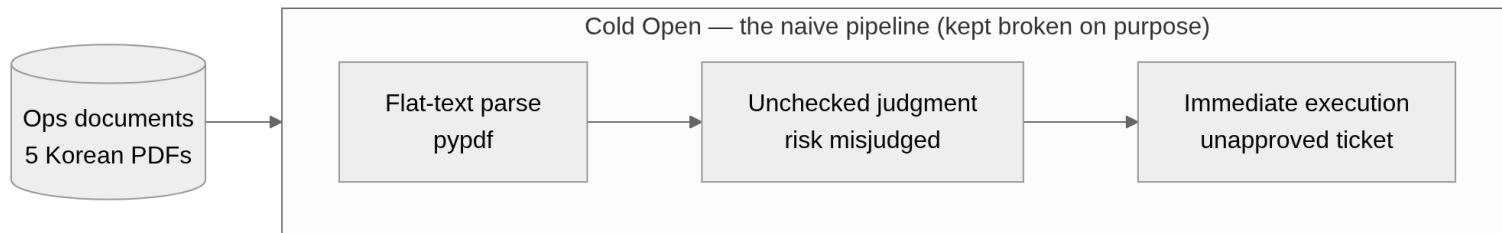
변경 작업에 사용되는 자동화 도구는 사전 등록된 도구 목록을 통해서만 호출할 수 있다. 미등록 도구 호출은 차단되며 감사 로그에 기록된다.

Demo



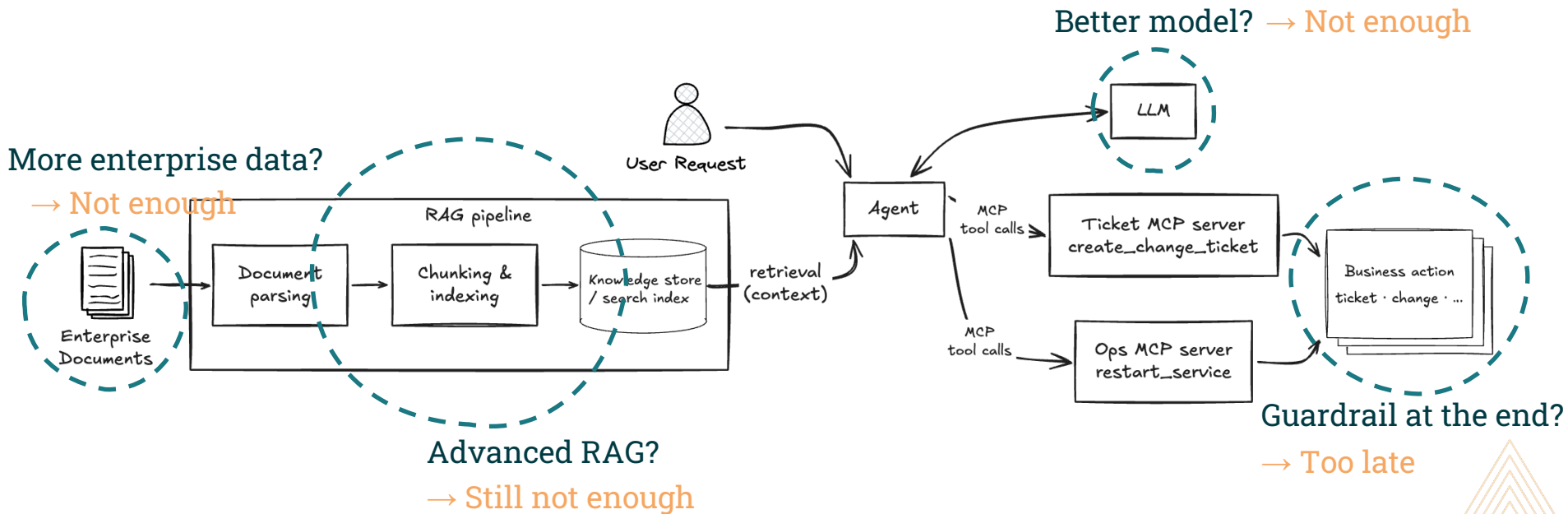
**No tool access control:
direct MCP connection, no audit trail**

4 Failed in Naïve Pipeline

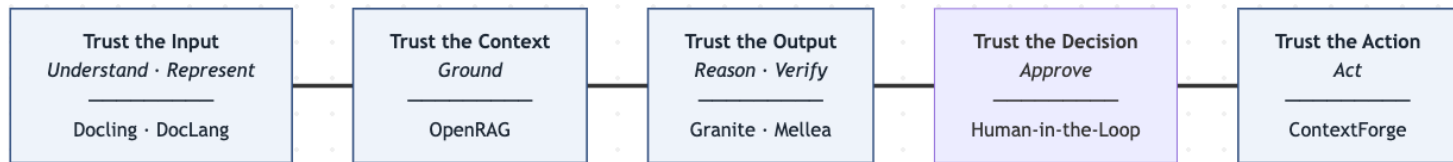


- ✓ Flat-text parsing lost the table structure → risk misjudged
- ✓ No validation: nobody checked the judgment against policy
- ✓ No approval: executed immediately
- ✓ Direct tool access: no control point, no audit trail

How can we fix it?



Composable Open Source Reference Architecture



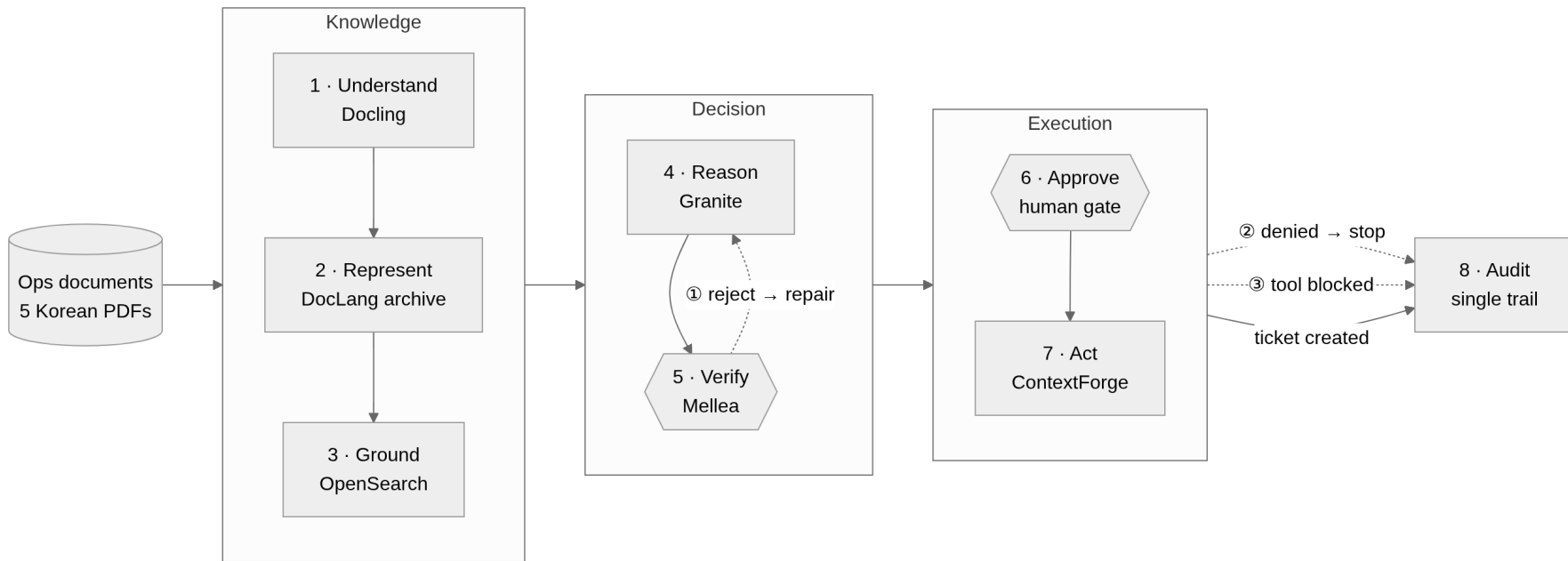
- ✓ Document intelligence with structure preservation
- ✓ Grounded retrieval with evidence tracing
- ✓ Output validation against business rules
- ✓ Policy enforcement and governance
- ✓ Auditable agent orchestration and tool control

Operational Benefits → Trust, Governance, Quality, Observability

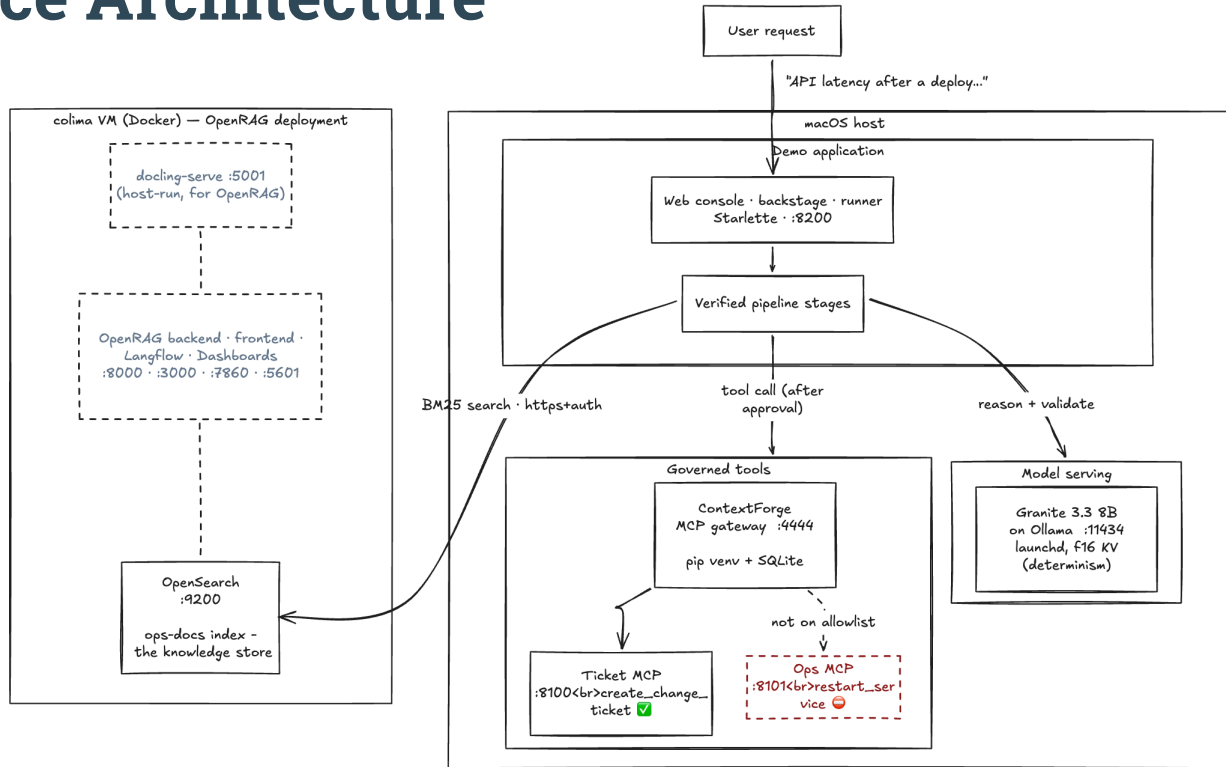


Trust Is a Chain

*“Trust controls are embedded **at every stage** instead of added as a final guardrail.”*

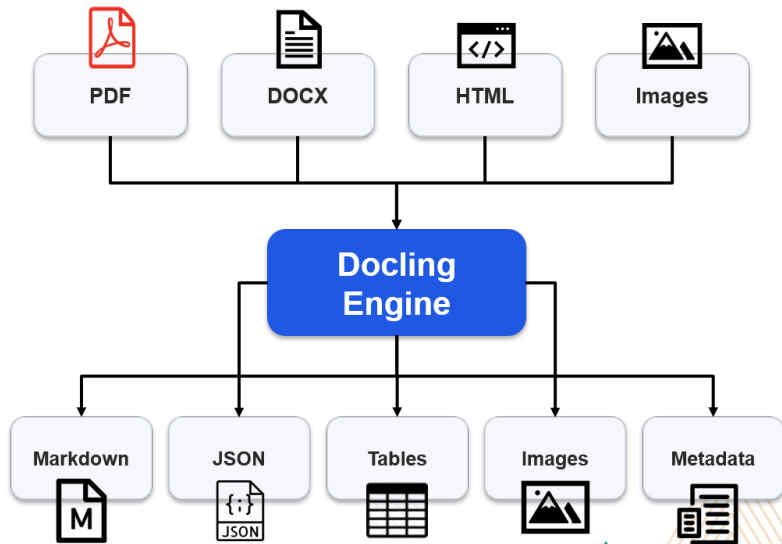


Reference Architecture



1 Docling – Document understanding Pipeline

- ✓ Converts PDFs, DOCX, HTML, PPTX, and scanned images into clean structured data.
- ✓ Preserves document meaning through layout analysis, table detection, figure extraction, and metadata.
- ✓ Outputs Markdown, JSON, tables, images, and metadata for downstream RAG pipelines.
- ✓ Solves the “first-mile” problem before retrieval, reasoning, and agent workflows.



Introducing Docling

: Community Adoption

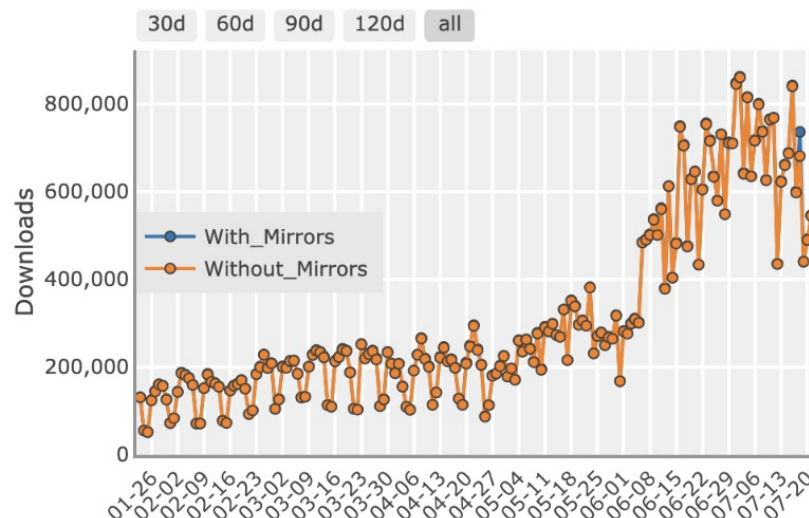
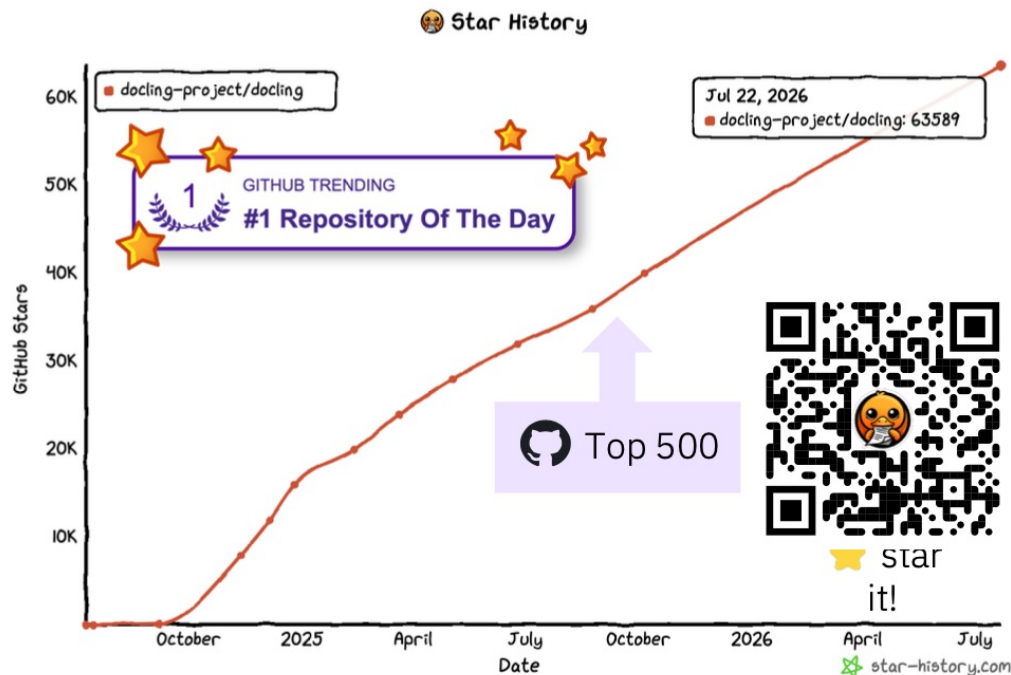
A rising star in the open-source space

★ 64k+ GitHub stars

📦 20M+ downloads/month from [PyPi](#)

+800k/day

Used by 2.4k



Introducing Docling

: Recovering structured content from PDF w/ low-level PDF parsers

! undesired
page headers

KDD '22, August 14–18, 2022, Washington, DC, USA Birgit Pfitzmann, Christoph Auer, Michele Dolfini, Nassar, and Peter Staar

Table 1: DocLayNet dataset overview. Along with the frequency of each class label, we present the relative occurrence (as % of row "Total") in the train, test and validation sets. The inter-annotator agreement is computed as the mAP@0.5-0.95 metric between pairwise annotations from the triple-annotated pages, from which we obtain accuracy ranges.

class label	Count	% of Total	Train	Test	Val	triplet inter-annotator mAP @ 0.5-0.95 (%)
Caption	22524	2.04	1.77	2.32	84.89	80-81 86-92 94-99 95-99 69-78 n/a
Footnote	6318	0.60	0.51	0.58	83.91	80-81 80-82 82-84 85-94 n/a 82-97
Formula	25027	2.25	1.90	2.96	83-85	80-81 n/a 84-87 86-96 n/a n/a
List-item	185660	17.19	13.34	13.82	87-88	74-83 90-92 97-97 81-83 75-88 93-95
Page-footer	70878	6.51	5.58	6.60	93-94	88-90 95-96 100 92-97 100 96-98
Page-header	58022	5.10	6.70	5.06	85-89	66-76 90-94 98-100 91-92 97-99 81-86
Picture	40576	3.71	2.78	5.31	48-71	56-59 82-86 49-82 80-93 66-71 59-76
Section-header	142884	13.00	13.77	12.81	83-84	70-81 90-92 94-95 87-94 49-73 78-86
Table	34733	3.20	2.27	3.60	77-81	75-80 83-86 98-99 58-80 78-84 70-83
Text	510377	45.82	49.18	45.00	84-86	81-86 48-61 89-93 87-92 71-79 87-95
Title	5071	0.47	0.30	0.50	68-72	28-63 30-63 96-100 82-96 68-79 28-56
Total	1107470	9611.23	99816	66531	82-83	71-74 79-81 89-94 86-91 71-76 68-85



Figure 3: Corpus Conversion Service annotation user interface. The PDF page is shown in the background, with overlaid text-cells (in darker shades). The annotation boxes can be drawn by dragging a rectangle over each segment with the respective label from the palette on the right.

we distributed the annotation workload and performed continuous quality controls. Phase one and two required a small team of experts only. For phases three and four, a group of 40 dedicated annotators were assembled and supervised.

Phase 1: Data selection and preparation. Our inclusion criteria for documents were described in Section 3. A large effort went into ensuring that all documents are free to use. The data sources

include publication repositories such as arXiv³, government offices, company websites as well as data directory services for financial reports and patents. Scanned documents were excluded wherever possible because they can be rotated or skewed. This would not allow us to perform annotations with rectangular bounding boxes and therefore complicate the annotation process.

Preparation work included uploading and parsing the scanned PDF documents in the Corpus Conversion Service (CCS) [22], a cloud-native platform which provides a visual annotation interface and allows for dataset inspection and analysis. The annotation interface of CCS is shown in Figure 3. The desired balance of pages between the different document categories was achieved by selective subsampling of pages with certain desired properties. For example, we made sure to include the title page of each document and bias the remaining page selection to those with figures or tables. The latter was achieved by leveraging pre-trained object detection models from PubLayNet, which helped us estimate how many figures and tables a given page contains.

Phase 2: Label selection and guidelines. We reviewed the collected documents and identified the most common structural features they exhibit. This was achieved by identifying recurrent layout elements and lead us to the definition of 11 distinct class labels. These 11 class labels are Caption, Footnote, Formula, List-item, Page-footer, Page-header, Picture, Section-header, Table, Text, and Title. Critical factors that were considered for the choice of these class labels were (i) the overall occurrence of the label, (ii) the specificity of the label, (iii) recognisability on a single page (i.e. no need for content from previous or next page) and (iv) overall coverage of the page. Specifically, ensuring that the choice of label is not ambiguous, while coverage ensures that all meaningful items on a page can be annotated. We refrained from class labels that are very specific to a document category, such as Abstract in the Scientific Journals category. We also avoided class labels that are tightly linked to the semantics of the text. Labels such as Author and Affiliation, as seen in DocBank, are often only distinguishable by discriminating on

³<https://arxiv.org/>

KDD '22, August 14–18, 2022, Washington, DC, USA Birgit Pfitzmann, Christoph Auer, Michele Dolfini, Nassar, and Peter Staar

Table 1: DocLayNet dataset overview. Along with the frequency of each class label, we present the relative occurrence (as % of row "Total") in the train, test and validation sets. The inter-annotator agreement is computed as the mAP@0.5-0.95 metric between pairwise annotations from the triple-annotated pages, from which we obtain accuracy ranges.

% of Total

triplet inter-annotator mAP @ 0.5-0.95 (%)

[...]
Count
22524
6318
25027
185660
70878
58022
45976
142884
34733
510377
5071
1107470

! Tables not
understood

! Image content
missing

[...]

include publication repositories such as arXiv³, government offices, company websites as well as data directory services for financial reports and patents. Scanned documents were excluded wherever possible because they can be rotated or skewed. This would not allow us to perform annotation with rectangular bounding-boxes and therefore complicate the annotation process.

[...]
! Multi-column often
breaks order

! Line wraps not
understood

✓ Very fast and cheap

✗ Incomplete

✗ Loss of structure

✗ Noisy

➡ Unfit for most use cases

Introducing Docling

: Schema-based information extraction

Extracting specific pieces of information from a document



hello

EVENTURE
EVENT PLANNER

THIS IS YOUR INVOICE

BILLED TO: Jonathan Patterson
PAY TO: Eventure & Co.
Bank: Borsche Bank
Account No: 0123 4567 8901

DESCRIPTION	UNIT PRICE	QTY	TOTAL
Flower Arrangements	100	1	\$100
Balloon Planning	100	1	\$100
Wedding Hall Decor	100	1	\$100
Reception Planning	100	1	\$100
Concierge	100	1	\$100
SUBTOTAL			\$500
		Tax	10%
		TOTAL	\$550

INVOICE NO: 01234
DATE: 11.09.2020
DUE DATE: 11.09.2020

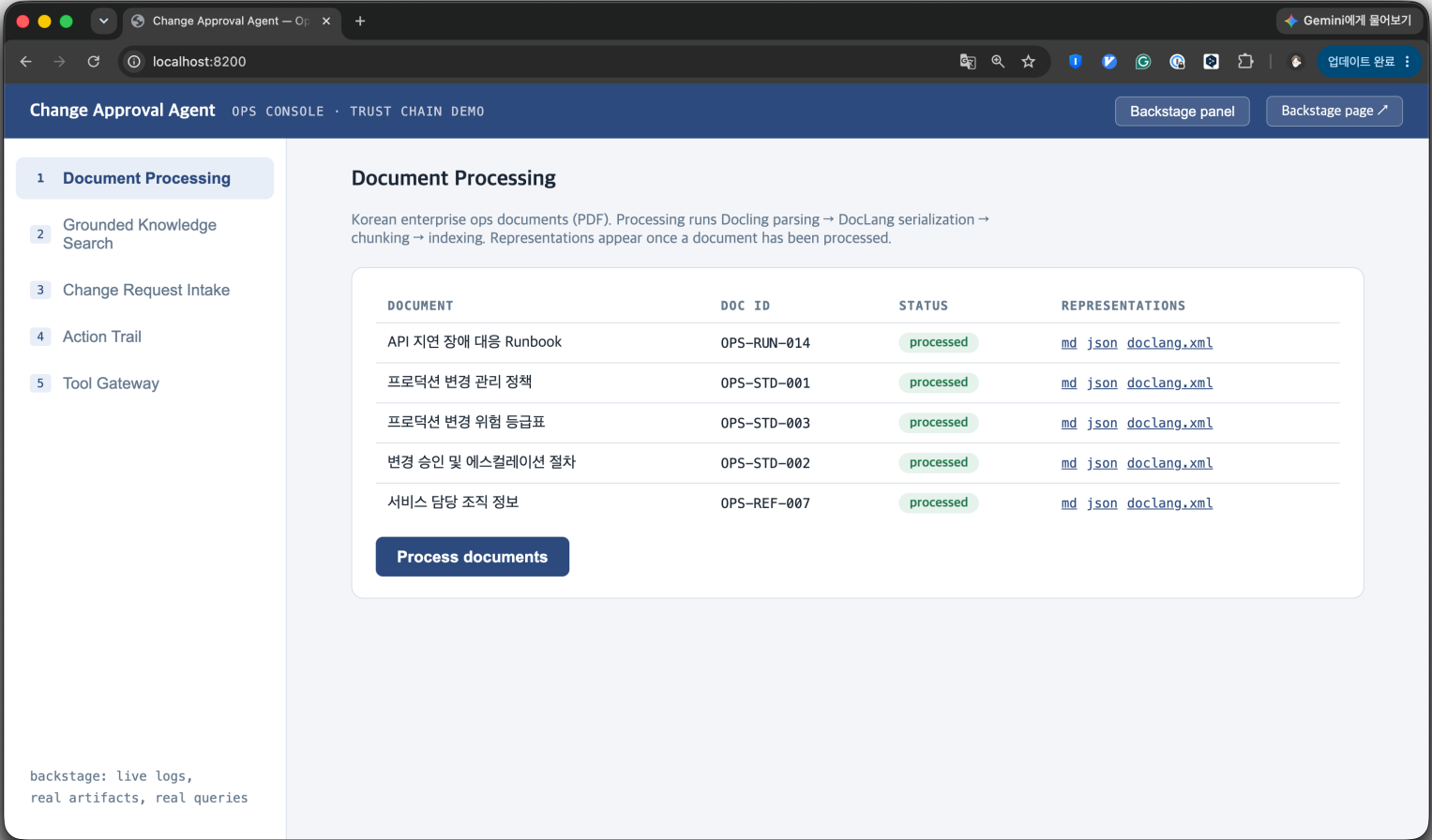
THANK YOU

+



```
{  
  'bill number': '01234',  
  'total invoice price': 550,  
  'currency of total invoice price': 'USD',  
  'name of invoice addressee': 'Jonathan Patterson',  
  'name of invoice sender': 'Eventure Event Planner'  
}
```

```
invoice_dict = {  
  "bill number": "string",  
  "total invoice price": "float",  
  "currency of total invoice price": "string",  
  "name of invoice addressee": "string",  
  "name of invoice sender": "string",  
}
```



프로덕션 변경 위험 등급표

문서번호: OPS-STD-003 · 개정: v2.4 (2026-05-02) · 관리부서: 플랫폼운영팀 · 등급: 사내한

본 문서는 프로덕션 환경 변경 요청(RFC)의 위험 등급 판정 기준을 정의한다. 변경 요청자는 아래 등급표에 따라 위험 등급을 판정하고, 등급별 필수 요건을 충족한 후 변경을 진행해야 한다.

1. 위험 등급 판정 기준

위험 등급	판정 기준		필수 요건	
	영향 범위	변경 유형	승인 주체	통백 계획
High	프로덕션 전체	데이터베이스 스키마 마이그레이션	CAB(변경자문위원회)	필수 ※2
	프로덕션 전체	API 인터페이스 변경		
	다수 서비스	공유 인프라 구성 변경		
Medium	단일 서비스	애플리케이션 재배포	서비스 오퍼	필수
	단일 서비스	런타임 설정 값 변경		권장
	내부 도구	문서/대시보드 변경	팀 리더	불필요
Low	스테이징 환경	모든 유형	자동 승인	

※1 프로덕션 장애 발생 이후 해당 장애에 대한 후속 변경(긴급 수정, 설정 변경, 재배포 포함)은 본 표의 판정 결과에서 한 등급 상향하여 적용한다.

※2 위험 등급 High의 변경은 승인 완료 전 어떠한 변경 작업(도구 실행 포함)도 착수할 수 없다.

※3 등급 판정이 모호한 경우 상위 등급을 적용한다.

2. 등급별 처리 시한

위험 등급	승인 처리 시한	변경 가능 시간대
High	영업일 기준 2일	정기 점검 시간대(일 02:00-05:00)
Medium	영업일 기준 1일	업무 시간 외
Low	즉시	제한 없음

Naïve Processing

```
1 ## 프로덕션 변경 위험 등급표
2
3 문서번호: OPS-STD-003 · 개정: v2.4 (2026-05-02) · 관리부서: 플랫폼운영팀 · 등급: 사내한
4
5 본 문서는 프로덕션 환경 변경 요청(RFC)의 위험 등급 판정 기준을 정의한다. 변경 요청자는 아래 등급표에 따라 위험 등급을 판정하고, 등급별 필수 요건을 충족한 후 변경을 진행해야 한다.
6
7
8
```

1. 위험 등급 판정 기준

위험 등급	판정 기준	판정 기준	필수 요건	필수 요건
위험 등급	영향 범위	변경 유형	승인 주체	통백 계획
High	프로덕션 전체	데이터베이스 스키마 마이그레이션	CAB(변경자문위원회)	필수 ※2
High	프로덕션 전체	API 인터페이스 변경	CAB(변경자문위원회)	필수 ※2
High	다수 서비스	공유 인프라 구성 변경	CAB(변경자문위원회)	필수 ※2
Medium	단일 서비스	애플리케이션 재배포	서비스 오퍼	필수
Medium	단일 서비스	런타임 설정 값 변경	서비스 오퍼	권장
Low	내부 도구	문서/대시보드 변경	팀 리더	불필요
Low	스테이징 환경	모든 유형	자동 승인	불필요

※1 프로덕션 장애 발생 이후 해당 장애에 대한 후속 변경(긴급 수정, 설정 변경, 재배포 포함)은 본 표의 판정 결과에서 한 등급 상향 하여 적용한다.

※2 위험 등급 High의 변경은 승인 완료 전 어떠한 변경 작업(도구 실행 포함)도 착수할 수 없다.

※3 등급 판정이 모호한 경우 상위 등급을 적용한다.

2. 등급별 처리 시한

위험 등급	승인 처리 시한	변경 가능 시간대
High	영업일 기준 2일	정기 점검 시간대(일 02:00:00)
Medium	영업일 기준 1일	업무 시간 외
Low	즉시	제한 없음

Docling

```
343 {
344     "bbox": {
345         "l": 231.23436536523437,
346         "t": 189.9329917202749,
347         "r": 264.7413339083606,
348         "b": 198.00599178754987,
349         "coord_origin": "TOPLEFT"
350     },
351     "row_span": 1,
352     "col_span": 2,
353     "start_row_offset_idx": 0,
354     "end_row_offset_idx": 1,
355     "start_col_offset_idx": 1,
356     "end_col_offset_idx": 3,
357     "text": "판정 기준",
358     "column_header": true,
359     "row_header": false,
360     "row_section": false,
361     "fillable": false
362 },
```

² DocLang – AI Native Document Format

- ✓ Open markup standard for representing documents in an **AI-readable format**
- ✓ Preserves **structure, semantics, layout, and relationships**
- ✓ Optimized for **LLM and agent consumption** with minimal information loss
- ✓ Complements **Docling** by standardizing parsed document output

• WHAT YOUR PARSER RETURNS

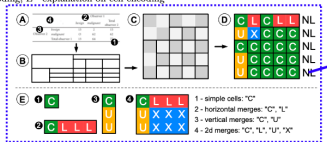
```
Q3 2024Financial Re
port Net Revenue42
M51M39M Figure3.2
omitted author:null
```

• WHAT DOCLANG RETURNS

```
<head><author>J. Smith</author></head>
<heading level="1">Q3 2024</heading>
<table><thead><tr><th>Revenue</th><th>Figure</th></tr></thead>
<tbody><tr><td>42M</td><td>3.2</td></tr></tbody></table>
<figure></figure>
```

Introducing DocLang

Fig. 3. OTSL description of table structure: A - table example; B - graphical representation of table structure; C - mapping structure on a grid; D - OTSL structure encoding; E - explanation on cell encoding



1 - simple cells: "C"
2 - horizontal merges: "C", "L"
3 - vertical merges: "C", "U"
4 - 2d merges: "C", "U", "L", "X"

Table 2. TSR and cell detection results compared between OTSL and HTML on the PubTabNet [22], FinTabNet [21] and PubTables-1M [14] data sets using TableFormer [9] (with enc=6, dec=6, heads=8).

Data set	Language	TEDs			mAP(0.75)	Inference time (secs)
		simple	complex	all		
PubTabNet	OTSL	0.965	0.934	0.955	0.88	2.73
	HTML	0.969	0.927	0.955	0.857	5.39
FinTabNet	OTSL	0.955	0.961	0.959	0.862	1.85
	HTML	0.917	0.922	0.92	0.722	3.26
PubTables-1M	OTSL	0.987	0.964	0.977	0.896	1.79
	HTML	0.983	0.944	0.966	0.889	3.26

5.3 Qualitative Results

Figure 6: Qualitative differences between OTSL and HTML. Figure 6 shows only 5 tokens that directly describe a tabular structure based on an atomic 2D grid. The OTSL vocabulary is comprised of the following tokens:

- "C" cell - a new table cell that either has or does not have cell content
- "L" cell - left-looking cell, merging with the left neighbor cell to create a span
- "U" cell - up-looking cell, merging with the upper neighbor cell to create a span
- "X" cell - cross cell, to merge with both left and upper neighbor cells
- "NL" - new-line, switch to the next row.

A notable attribute of OTSL is that it has the capability of achieving lossless HTML.

An expression using a data frame called `df` in R where you want to summarize `x` by month:

```
require(plyr)
df <- data.frame(
  x = runif(120, 1, 160),
  y = runif(120, 7, 334),
  z = runif(120, 1.7, 20.7),
  month = rep(c(5,6,7,8),30),
  week = sample(1:4, 120, TRUE)
)

ddply(df, .(month, week), summarize,
  mean = round(mean(x), 2),
  sd = round(sd(x), 2))
```

```
<picture>
  <label value="flow_chart"/>
  <location value="102"/><location value="85"/><location value="416"/><location value="160"/>
  <caption>Fig. 3. OTSL description of table structure: A - table example; B - graphical ... cell encoding</caption>
</picture>

<table>
  <location value="63"/><location value="78"/><location value="436"/><location value="158"/>
  <caption>Table 2. ISR and cell detection results compared between OTSL and ... heads=8)</caption>
  <ched/>Data set<ched/>Lang<ched/>TEDs<lccl/><lccl/><ched/>MAP (0.75)<ched/>Inference time (secs)<nl/>
  <ucel/><ucel/><ched/>simple<ched/>complex<ched/>all<ucel/><ucel/><nl/>
  <ched/>PubTabNet<ched/>OTSL<fccl/>0.965<fccl/>0.934<fccl/>0.955<fccl/>0.88<fccl/>2.73<nl/>
  <ucel/><ched/>HTML<fccl/>0.969<fccl/>0.927<fccl/>0.955<fccl/>0.857<fccl/>5.39<nl/>
  <!--...-->
</table>

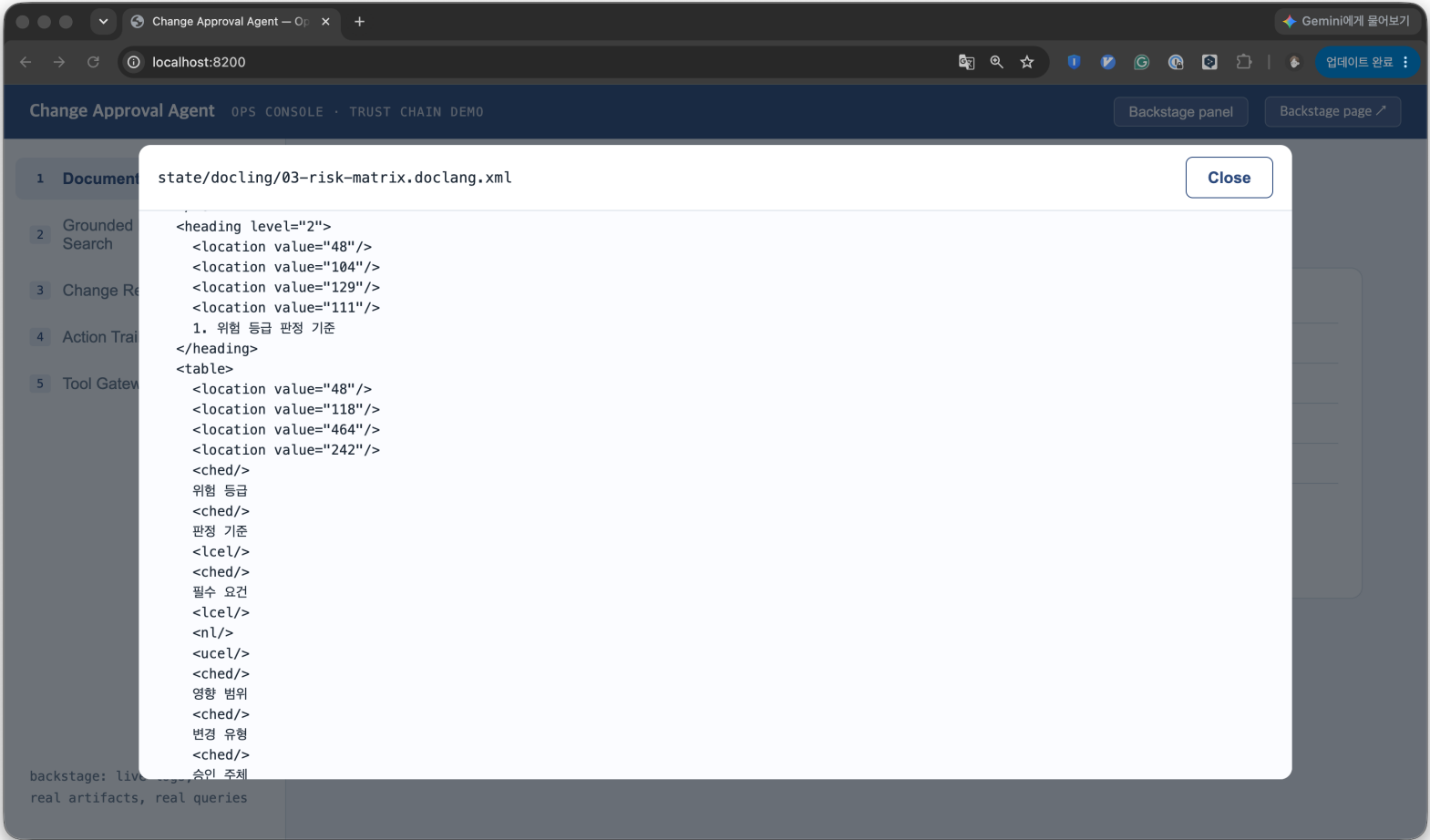
<heading level="2"><location value="58"/><location value="343"/><location value="209"/><location value="351"/>
  5.2 Quantitative Results</heading>

<text><location value="66"/><location value="360"/><location value="358"/><location value="368"/>The OTSL ... tokens:</text>
<list>
  <ldiv/><location value="53"/><location value="377"/><location value="439"/><location value="385"/>"C" cell ...
  <ldiv/><location value="53"/><location value="387"/><location value="439"/><location value="403"/>"L" cell ...
  <ldiv/><location value="53"/><location value="405"/><location value="439"/><location value="421"/>"U" cell ...
  <ldiv/><location value="53"/><location value="423"/><location value="409"/><location value="431"/>"X" cell ...
  <ldiv/><location value="53"/><location value="433"/><location value="269"/><location value="441"/>"NL" - ...
</list>

<code>
  <label value="R"/>
  <location value="58"/><location value="170"/><location value="443"/><location value="266"/>
  <content><!CDATA[require (plyr)
df <- data.frame(
  x = runif(120, 1, 160),
  y = runif(120, 7, 334),
  z = runif(120, 1.7, 20.7),
  month = rep(c(5,6,7,8), 30),
  week = sample(1:4, 120, TRUE)
)

ddply(df, .(month, week), summarize,
  mean = round(mean(x), 2),
  sd = round(sd(x), 2)]]]></content>
</code>
```

Demo



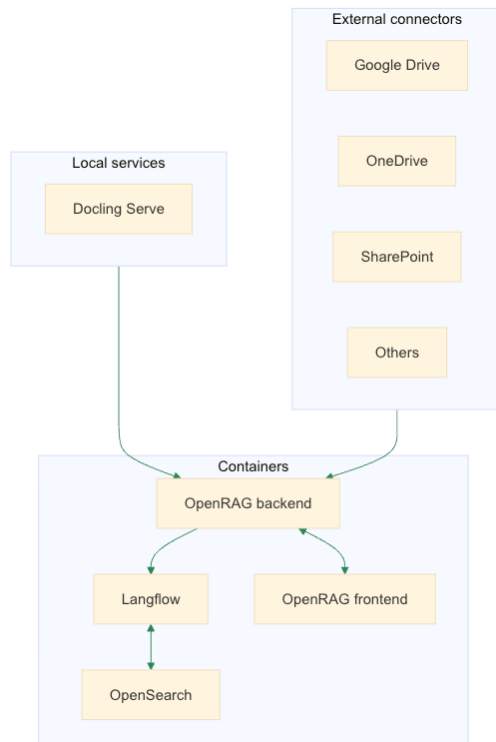
Docling Is the Translator, DocLang Is the Language

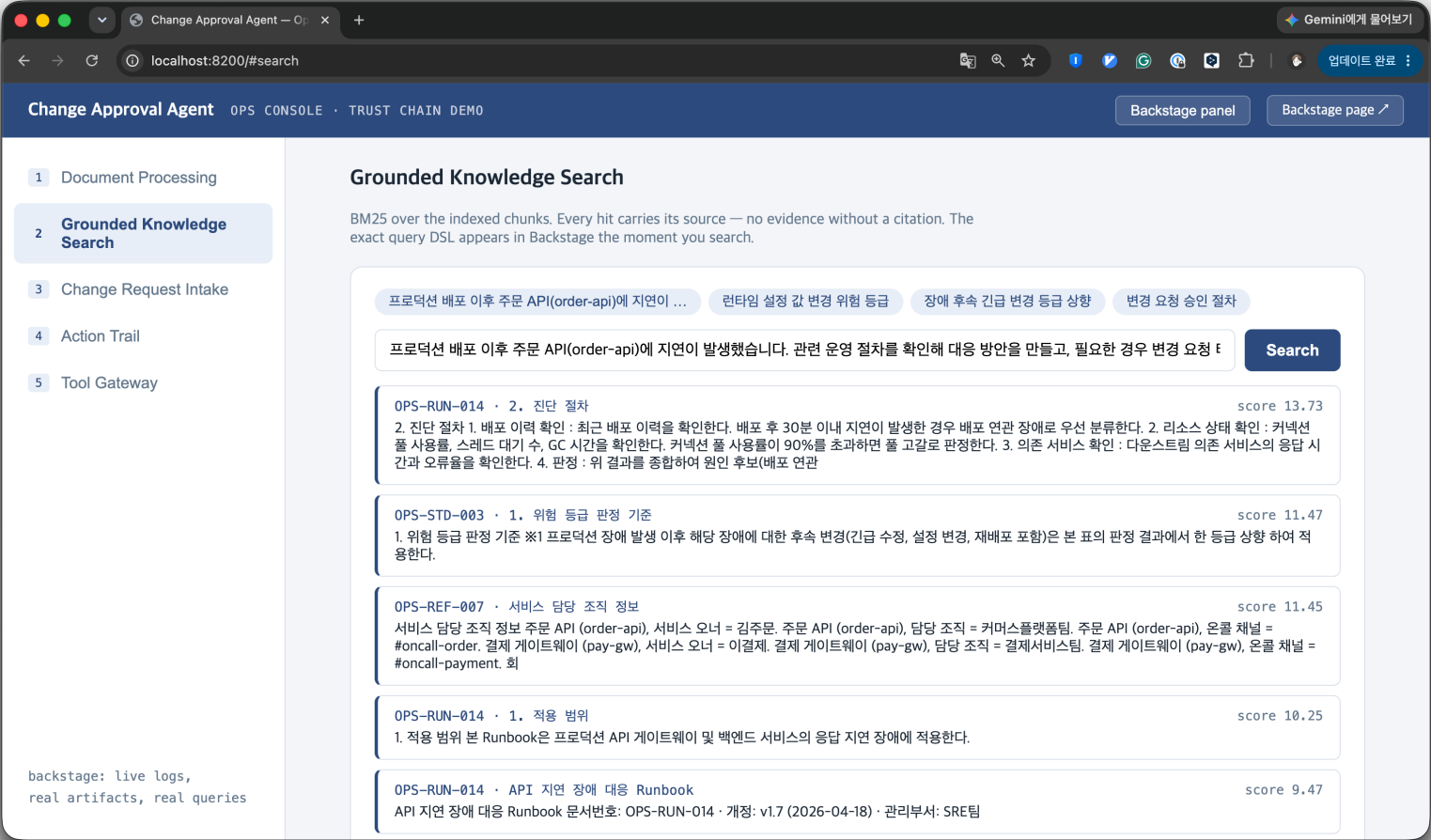
- ✓ Preserve meaning, relationships, and business context
- ✓ Reduce errors, manual review, and time-to-decision
- ✓ Carry compliance metadata and rules with the document
- ✓ Enable portable, vendor-neutral AI workflows



3 OpenRAG – Composable Open-Source RAG Platform

- ✓ Building Agentic RAG applications quickly and flexibly
- ✓ Integrates Docling, OpenSearch, and Langflow into a unified RAG stack
- ✓ Supports the end-to-end pipeline (document parsing, chunking, retrieval, workflow orchestration)
- ✓ Provides a composable architecture to easily replace or extend models, retrieval components, tools, APIs, and MCP integrations





4 IBM Granite

✓ **Greater Efficiency & Lower Costs**

Granite's compact, efficient models reduce infrastructure costs and compute demands while accelerating AI workloads.

✓ **Security & Safety**

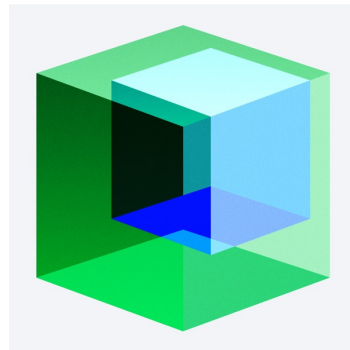
Built with enterprise-grade transparency, security, and governance to support responsible AI deployment.

✓ **Flexibility & Scalability**

Deploy Granite across environments and scale efficiently for large datasets and complex workloads.

✓ **Customizability**

Adapt Granite with your data for high performance at lower cost.



4 IBM Granite

Open Performant Trusted



ISO 42001 Certification

A comprehensive framework for AI Management Systems, ensuring responsible development and deployment of AI through ethics, accountability, and trust.



IBM Granite models are the first open-source AI models to achieve ISO 42001 certification



Enables confident deployment of Granite models in highly regulated industries which prioritize compliance



Achieved in < 3 months and with zero non-conformities, confirming IBM's mature AI governance standards



⁵ Mellea – Reliable, Testable Generative AI

- ✓ **Structured generative programming**

Build AI applications around explicit instructions and requirements, not just prompts

- ✓ **Instruct → Validate → Repair**

Validate generated outputs and automatically repair results that fail requirements

- ✓ **Requirements as guardrails**

Enforce business rules, formats, and constraints through explicit validation

- ✓ **Reliable and controllable outputs**

Make generative AI more predictable, testable, and production-ready



Prompt engineering



Mellea abstractions:

- Safety guardrails
- Instructions
- Requirements
- Sampling Strategies



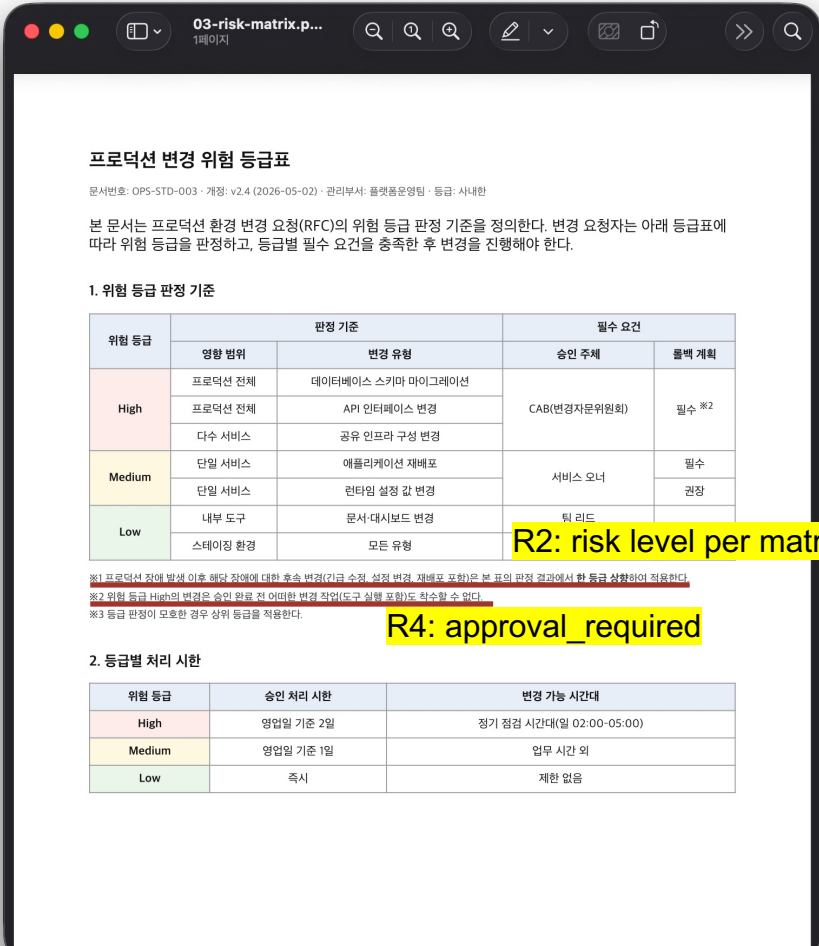
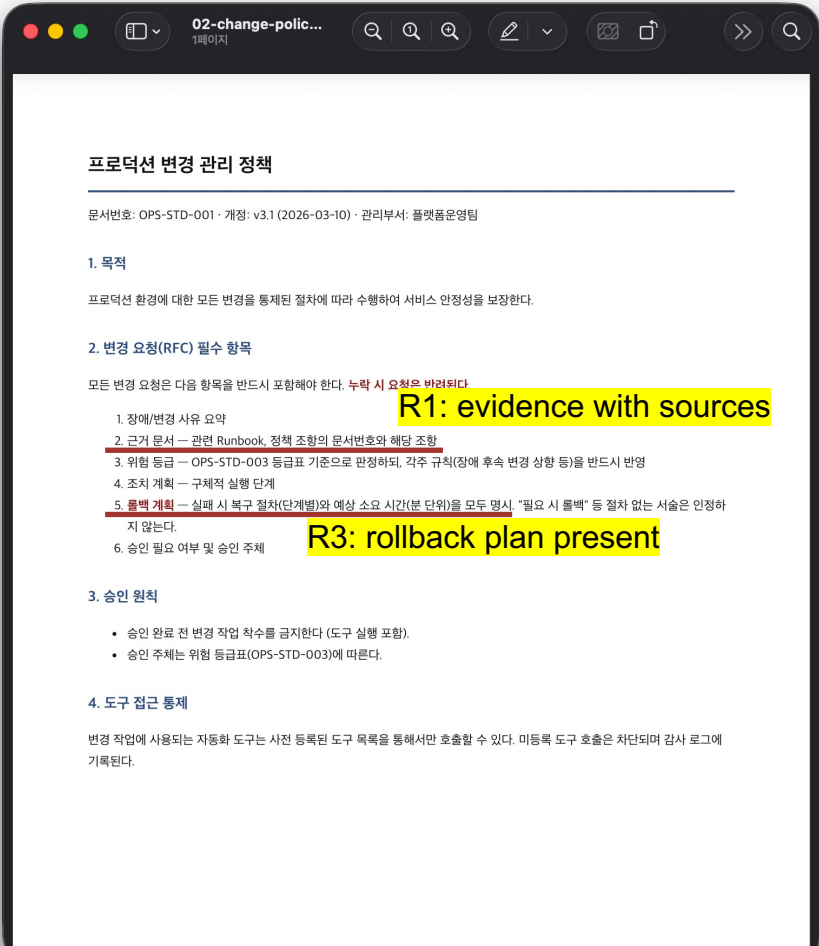
Generative computing

→ LLM Generative computing

```
# Step 1: Screen inputs for attacks
Safety_Issue_Flag = backend_guardian.generate("{{context}}") == "Yes"
if Safety_Issue_Flag:
    print("ALERT: Adversarial Attack Detected")
else:

# Step 2: Create Outline
outline_instruction = "Create an outline for a report on how {{current_subtopic}} \
    impacts {{main_topic}}. You have the following research available:\n{{research}}"
req_markdown = Requirement("Use markdown syntax.", fn=check_markdown)
req_num_sections = Requirement("Limit the number of subsections to \
    a maximum of {{max_subsections}}.", fn=check_section_count)
req_outline = Requirement("Do not include these standard sections: \
    # Introduction, # Conclusion, # References", fn=check_sections)
outline = backend.sample_against_requirements(
    outline_instruction,
    requirements = [req_outline, req_markdown, req_num_sections],
    user_variables = user_args
    budget = 5
)

# Step 3: Generate Report
report_instruction = "Context:\n{{context}}\nSummarize the relevant information \
    available into a detailed report on how {{current_subtopic}} impacts \
    {{main_topic}}.\n\nFollow this outline:\n{{outline}}",
req_focus = Requirement("Stay focused on the topic, avoid unrelated information.")
req_language = Requirement("Write the report in {{language}}.")
req_tone = Requirement("Use an {{tone}} tone throughout the report.")
req_length = Requirement("The report should have a minimum length \
    of {{total_words}} words.", fn=check_word_count)
report = backend.generate_with_repair(
    outline_instruction,
    requirements = [req_focus, req_language, req_tone, req_length],
    user_variables = user_args
)
```



Change Approval Agent — O: X +

localhost:8200/#intake

Gemini에게 물어보기

← → ↺

📄 🔍 ☆

📄 📱 📧 📧 📧 📧 📧

업데이트 완료

Change Approval Agent

OPS CONSOLE · TRUST CHAIN DEMO

Backstage panel

Backstage page ↗

1 Document Processing

2 Grounded Knowledge Search

3 Change Request Intake

4 Action Trail

5 Tool Gateway

backstage: live logs,
real artifacts, real queries

Change Request Intake

The demo scenario request (fixed — the whole talk uses this one question). The agent grounds it, drafts a plan, and Mellea validates before anything can happen.

프로덕션 배포 이후 주문 API(order-api)에 지연이 발생했습니다. 관련 운영 절차를 확인해 대응 방안을 만들고, 필요한 경우 변경 요청 티켓을 생성해 주세요.

Run agent

Validation history — 2 draft(s) rejected by Mellea before this plan was accepted. Click an attempt for details:

#1 DraftChangePlan rejected: R2, R3

#2 ChangePlan rejected: R1

#3 ChangePlan all requirements passed

Validated ChangePlan — waiting for a human decision

프로덕션 배포 후 주문 API(order-api)에서 지연이 발생했습니다.

RISK LEVEL
High

ROLLBACK STEPS
2

EVIDENCE
1 sources

APPROVAL REQUIRED
yes

Instruct – Validate – Repair

Requirement:

- Generation
- Validation



```
plan = m.instruct(  
    "Draft a ChangePlan, grounded in the retrieved policy. ...",  
    requirements = ["A rollback plan must be present."],  
)
```

Instruct – Validate – Repair

Default validation
strategy is
LLM-as-a-Judge



```
requirements = [  
    req("Every claim must cite a source document"),  
    req("A rollback plan must be present",  
        validation_fn = simple_validate(has_rollback_plan)),  
    req("A High-risk change must require approval"),  
]  
  
plan = m.instruct(  
    "Draft a ChangePlan, grounded in the retrieved policy. ...",  
    requirements = requirements,  
    format = ChangePlan,  
)
```

Instruct – Validate – Repair

But provide your
Own logic when
possible!



```
requirements = [  
    req("Every claim must cite a source document"),  
    req("A rollback plan must be present",  
        validation_fn = simple_validate(has_rollback_plan)),  
    req("A High-risk change must require approval"),  
]  
  
def has_rollback_plan(output: str) -> bool:  
    plan = ChangePlan.model_validate_json(output)  
    return plan.rollback_plan is not None  
  
plan = m.instruct(  
    "Draft a ChangePlan, grounded in the retrieved policy. ...",  
    requirements = requirements,  
    format = ChangePlan,  
)
```

Instruct – Validate – Repair

If requirement
fails, feed the
failure back
and try again.
(But not forever!)



```
requirements = [...]  
  
plan = m.instruct(  
    "Draft a ChangePlan, grounded in the retrieved policy. ...",  
    requirements = requirements,  
    format = ChangePlan,  
    strategy=RepairTemplateStrategy(loop_budget=3),  
)
```

Change Approval Agent — OPS · x +

localhost:8200/#intake

Change Approval Agent OPS CONSOLE · TRUST CHAIN

1 Document

2 Grounded Search

3 Change R

4 Action Trai

5 Tool Gatew

IVR attempt #1 – response format:

Requirement checks:

PASS R1 evidence with sources

FAIL R2 risk level per matrix + ✖

FAIL R3 rollback plan present

PASS R4 approval required

Model output (verbatim):

{

"incident_summary": "주문 API(order-api)에 대한 프로덕션 배포 후 지연이 발생했습니다.",

"evidence": [

{

"doc_id": "[OPS-RUN-014 2. 진단 절차]",

"doc_title": "OPS-RUN-014 진단 절차",

"section": "1. 배포 이력 확인",

"quote": "최근 배포 이력을 확인하고, 우선 분류한다."

{

"doc_id": "[OPS-RUN-014 2. 진단 절차]",

"doc_title": "OPS-RUN-014 진단 절차",

"section": "2. 리소스 상태 확인",

"quote": "커넥션 풀 사용률이 90%를 초과하면 풀 고갈로 판정한다."

{

"doc_id": "[OPS-RUN-014 2. 진단 절차]",

"doc_title": "OPS-RUN-014 진단 절차",

"section": "3. 의존 서비스 확인",

"quote": "다운스트림 의존 서비스의 응답 시간과 오류율을 확인한다."

backstage: live, real artifacts, real queries

RISK LEVEL

High

```
granite3.3:8b drafting (response format: DraftChangePlan)...
Attempt 1 – Granite draft (hasty output contract)
{
  "incident_summary": "주문 API(order-api)에 대한 프로덕션 배포 후 지연이 발생했습니다.",
  "evidence": [
    {
      "doc_id": "[OPS-RUN-014 2. 진단 절차]",
      "doc_title": "OPS-RUN-014 진단 절차",
      "section": "1. 배포 이력 확인",
      "quote": "최근 배포 이력을 확인하고, 배포 후 30분 이내 지연이 발생한 경우 배포 관련 장애로 우선 분류한다."
    },
    {
      "doc_id": "[OPS-RUN-014 2. 진단 절차]",
      "doc_title": "OPS-RUN-014 진단 절차",
      "section": "2. 리소스 상태 확인",
      "quote": "커넥션 풀 사용률이 90%를 초과하면 풀 고갈로 판정한다."
    },
    {
      "doc_id": "[OPS-RUN-014 2. 진단 절차]",
      "doc_title": "OPS-RUN-014 진단 절차",
      "section": "3. 의존 서비스 확인",
      "quote": "다운스트림 의존 서비스의 응답 시간과 오류율을 확인한다."
    }
  ],
  "risk_level": "Medium",
  "recommended_actions": [
    "배포 관련 장애, 리소스 고갈 또는 의존 서비스 문제를 조사합니다.",
    "커넥션 풀 설정(최대 크기, 타임아웃)을 조정하고, 필요에 따라 변경 요청(RFC) 절차를 따릅니다. (OPS-RUN-014 3.2 런타임 설정 변경 참조)"
  ],
  "approval_required": true,
  "policy_references": [
    "[OPS-STD-003 1. 위험 등급 판정 기준]",
    "[OPS-RUN-014 1. 적용 범위]",
    "[OPS-RUN-014 API 지연 장애 대응 Runbook]"
  ]
}
```

PASS R1 evidence with sources
FAIL R2 risk level per matrix + ✖
FAIL R3 rollback plan present
PASS R4 approval required



Change Approval Agent — OPS CONSOLE · TR

localhost:8200/#intake

Change Approval Agent

1 Document

2 Grounded Search

3 Change R

4 Action Trail

5 Tool Gatew

IVR attempt #2 – respons

Requirement checks:
FAIL R1 evidence with s
PASS R2 risk level per
PASS R3 rollback plan p
PASS R4 approval requi

Model output (verbatim):
{
 "incident_summary": "프로덕션 배포 이후 주문 API(order-api)에 지연이 발생했습니다.",
 "evidence": [
 {
 "doc_id": "[OPS-RUN-014 2. 진단 절차] 1",
 "doc_title": "OPS-RUN-014 2. 진단 절차",
 "section": "",
 "quote": "배포 후 30분

backstage: live
real artifacts, real queries

```
granite3.3:8b repairing against the full ChangePlan contract...
Attempt 2 – repaired plan (full ChangePlan contract)
{
  "incident_summary": "프로덕션 배포 이후 주문 API(order-api)에 지연이 발생했습니다.",
  "evidence": [
    {
      "doc_id": "[OPS-RUN-014 2. 진단 절차] 1",
      "doc_title": "OPS-RUN-014 2. 진단 절차",
      "section": "",
      "quote": "배포 후 30분 이내 지연이 발생한 경우 배포 연관 장애로 우선 분류한다."
    }
  ],
  "risk_level": "High",
  "recommended_actions": [
    "1. 배포 이력 확인: 최근 배포 이력을 확인하고, 지연이 발생한 경우 배포 연관 장애로 우선 분류.",
    "2. 리소스 상태 확인: 커넥션 풀 사용률, 스레드 대기 수, GC 시간을 모니터링합니다. 커넥션 풀 사용률이 90%를 초과하면 고갈 판정.",
    "3. 의존 서비스 확인: 다운스트림 의존 서비스의 응답 시간과 오류율을 검사합니다.",
    "4. 판정: 위 결과를 종합하여 원인 후보(배포 연관 / 리소스 고갈 / 의존 서비스 장애)를 기록하고 대응 옵션을 선택."
  ],
  "approval_required": true,
  "policy_references": [
    "OPS-STD-003"
  ],
  "rollback_plan": {
    "steps": [
      "1. 배포 이력 확인",
      "2. 리소스 상태 모니터링 및 조정(커넥션 풀 최대 크기, 커넥션 타임아웃, 요청 큐 길이)",
      "3. 의존 서비스 확인 및 문제 해결"
    ]
  },
  "estimated_minutes": 60
}
}

FAIL R1 evidence with sources
PASS R2 risk level per matrix + ✖
PASS R3 rollback plan present
PASS R4 approval required
```

Change Approval Agent — OPS CONSOLE · TRUST CHAIN

1 Document

2 Grounded Search

3 Change Request

4 Action Trail

5 Tool Gateway

IVR attempt #3 – response format:

Requirement checks:
PASS R1 evidence with sources
PASS R2 risk level per matrix + ✖
PASS R3 rollback plan present
PASS R4 approval required

Model output (verbatim):
{
 "incident_summary": "프로덕션 배포 후",
 "evidence": [
 {
 "doc_id": "[OPS-RUN-014 2. 진단 절차] 1",
 "doc_title": "OPS-RUN-014 2. 진단 절차",
 "section": "배포 이력 확인",
 "quote": "최근 배포 이력을 확인하고, 우선 분류한다."
 }
],
 "risk_level": "High",
 "recommended_actions": [
 "1. 리소스 상태 확인: 커넥션 풀 사용률, 스레드 대기 수, GC 시간을 모니터링하고 필요에 따라 조정합니다.",
 "2. 의존 서비스 확인: 다운스트림 의존 서비스의 응답 시간과 오류율을 검토하여 문제를 식별합니다.",
 "3. 변경 요청 생성: OPS-STD-001 2장에 따라 변경 요청 티켓을 작성하고, 장애 요약, 근거 문서, 위험 등급, 조치 계획, 롤백 계획을 포함합니다.",
 "4. 승인 절차: OPS-STD-003에 따라 등급별 승인 주체에게 변경 요청을 제출하고 승인을 받습니다."
],
 "approval_required": true,
 "policy_references": [
 "OPS-RUN-014",
 "OPS-STD-001",
 "OPS-STD-002",
 "OPS-STD-003"
],
 "rollback_plan": {
 "steps": [
 "배포를 이전 상태로 롤백합니다.",
 "문제 해결 후 시스템을 모니터링하여 지속적인 문제가 없는지 확인합니다."
],
 "estimated_minutes": 60
 }
},
backstage: live, real artifacts, real queries
RISK LEVEL: High

```
Attempt 3 – repaired plan (full ChangePlan contract)

{
  "incident_summary": "프로덕션 배포 후 주문 API(order-api)에서 지연이 발생했습니다.",
  "evidence": [
    {
      "doc_id": "[OPS-RUN-014 2. 진단 절차] 1",
      "doc_title": "OPS-RUN-014 2. 진단 절차",
      "section": "배포 이력 확인",
      "quote": "최근 배포 이력을 확인하고, 배포 후 30분 이내 지연이 발생한 경우 배포 관련 장애로 우선 분류한다."
    }
  ],
  "risk_level": "High",
  "recommended_actions": [
    "1. 리소스 상태 확인: 커넥션 풀 사용률, 스레드 대기 수, GC 시간을 모니터링하고 필요에 따라 조정합니다.",
    "2. 의존 서비스 확인: 다운스트림 의존 서비스의 응답 시간과 오류율을 검토하여 문제를 식별합니다.",
    "3. 변경 요청 생성: OPS-STD-001 2장에 따라 변경 요청 티켓을 작성하고, 장애 요약, 근거 문서, 위험 등급, 조치 계획, 롤백 계획을 포함합니다.",
    "4. 승인 절차: OPS-STD-003에 따라 등급별 승인 주체에게 변경 요청을 제출하고 승인을 받습니다."
  ],
  "approval_required": true,
  "policy_references": [
    "OPS-RUN-014",
    "OPS-STD-001",
    "OPS-STD-002",
    "OPS-STD-003"
  ],
  "rollback_plan": {
    "steps": [
      "배포를 이전 상태로 롤백합니다.",
      "문제 해결 후 시스템을 모니터링하여 지속적인 문제가 없는지 확인합니다."
    ],
    "estimated_minutes": 60
  }
}
```

```
PASS R1 evidence with sources
PASS R2 risk level per matrix + ✖
PASS R3 rollback plan present
PASS R4 approval required
```

 Validated ChangePlan — waiting for a human decision

프로덕션 배포 후 주문 API(order-api)에서 지연이 발생했습니다.

RISK LEVEL High	ROLLBACK STEPS 2	EVIDENCE 1 sources	APPROVAL REQUIRED yes
--------------------	---------------------	-----------------------	--------------------------

All four requirements passed, but nothing has been executed — and nothing can be until a human approves. The button below opens the approval decision (refusal ② of 3: the approval gate).

Review & decide — Approve / Deny

Demo

Validated ChangePlan — was approved

프로덕션 배포 후 주문 API(order-api)에서 지연이 발생했습니다.

RISK LEVEL
High

All four requirements passed, but not the approval. The button below opens the approval dialog.

Review & decide — Approve

Validation history — 2 draft(s) rejected by Mollies before this plan was accepted. Click an attempt for details.

Approve this change plan?

프로덕션 배포 후 주문 API(order-api)에서 지연이 발생했습니다.
risk **High** · rollback 2 steps · 1 evidence sources

Until you decide, no state-changing tool can be called — the gateway client enforces it, not the UI.

Deny Approve

APPROVAL REQUIRED
yes

Action Trail

Every decision and every tool call — allowed or blocked — in order. This view exists because the gateway is the single choke point.

Change ticket CHG-20260805-002
risk Medium · approver (승인 절차 없음) · rollback plan attached

- 0 approval_requested risk_level=High
- 1 approval_denied by=플랫폼운영팀 온콜 (데모)

Action Trail

Every decision and every tool call — allowed or blocked — in order. This view exists because the gateway is the single choke point.

restart_service — BLOCKED by ContextForge
The runbook's tempting shortcut is not in the agent's allowlist. The attempt is audited, not executed.

Change ticket CHG-20260805-003
risk High · approver 플랫폼운영팀 온콜 (데모) · rollback plan attached

- 0 approval_requested risk_level=High
- 1 approval_denied by=플랫폼운영팀 온콜 (데모)
- 2 approval_requested risk_level=High
- 3 approval_granted by=플랫폼운영팀 온콜 (데모)
- 4 tool_call_blocked tool=restart_service · via=change-approval virtual server · gateway_says=Tool not found: restart_service
- 5 tool_call_allowed tool=ticket-server-create-change-ticket · via=change-approval virtual server
- 6 ticket_created ticket_id=CHG-20260805-003 · approver=플랫폼운영팀 온콜 (데모)

6 ContextForge – MCP Gateway

- ✓ **Unified Tool Access**

Connect MCP, A2A, and REST/gRPC services through a unified gateway for AI agents.

- ✓ **Security & Governance**

Enforce authentication, RBAC, policies, and audit trails across agent–tool interactions.

- ✓ **Discovery & Management**

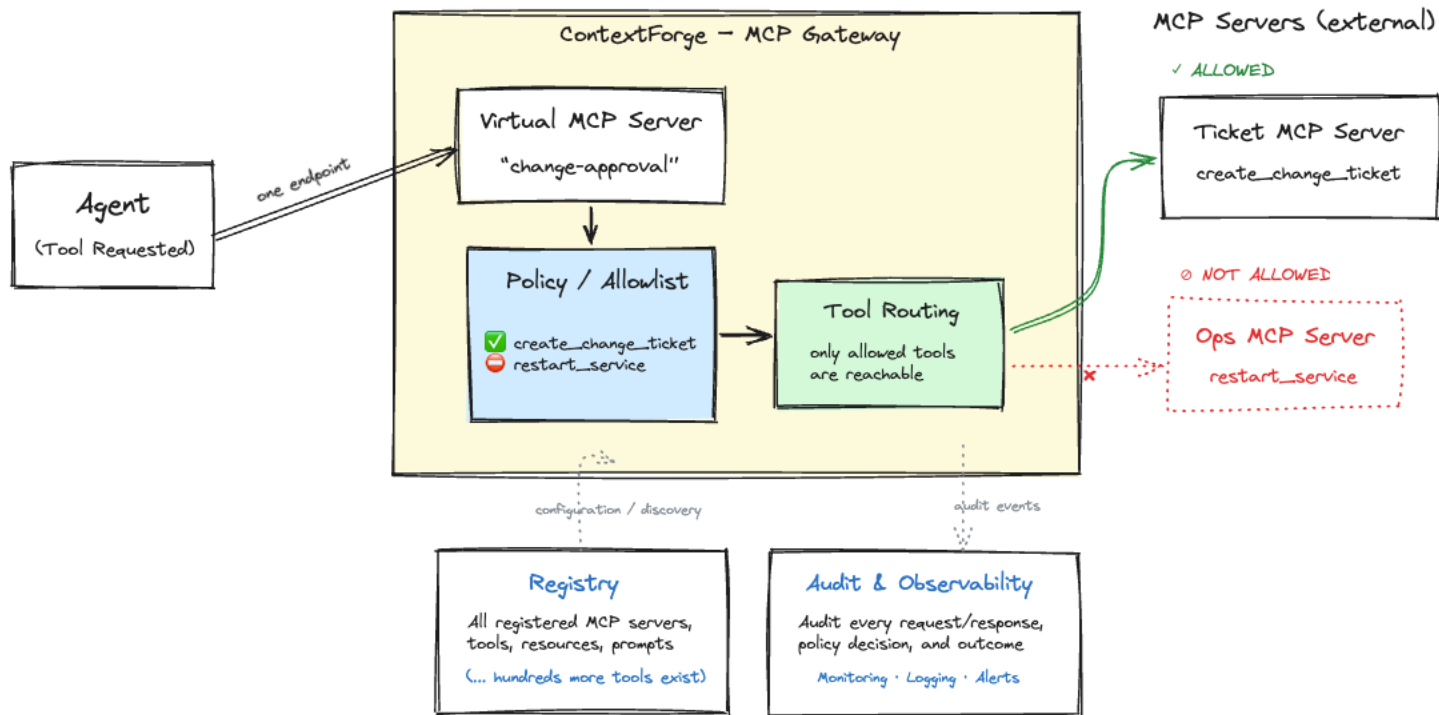
Centralize tool discovery, routing, and lifecycle management across distributed services.

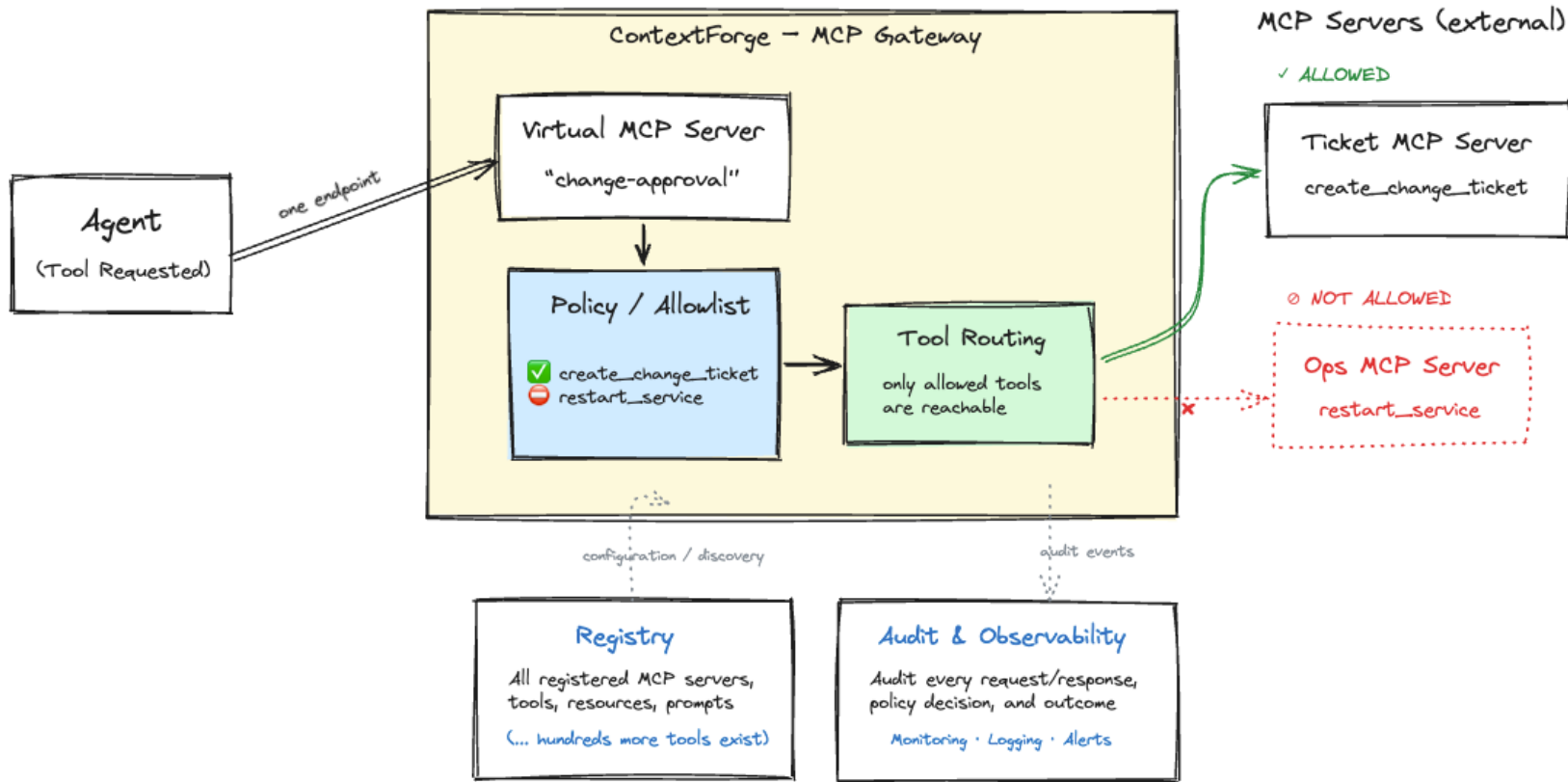
- ✓ **Scalability**

Monitor agent and tool activity with built-in observability while scaling for enterprise workloads.



6 ContextForge – MCP Gateway





Action Trail

Every decision and every tool call — allowed or blocked — in order. This view exists because the gateway is the single choke point.

⚠ restart_service — BLOCKED by ContextForge

The runbook's tempting shortcut is not in the agent's allowlist. The attempt is audited, not executed.

✅ Change ticket CHG-20260805-003

risk High · approver 플랫폼운영팀 온콜 (데모) · rollback plan attached

- 0 approval_requested risk_level=High
- 1 approval_denied by=플랫폼운영팀 온콜 (데모)
- 2 approval_requested risk_level=High
- 3 approval_granted by=플랫폼운영팀 온콜 (데모)
- 4 tool_call_blocked tool=restart_service · via=change-approval virtual server · gateway.says=Tool not found: restart_service
- 5 tool_call_allowed tool=ticket-server-create-change-ticket · via=change-approval virtual server
- 6 ticket_created ticket_id=CHG-20260805-003 · approver=플랫폼운영팀 온콜 (데모)

Tool Gateway — ContextForge

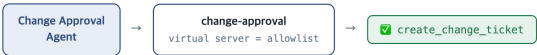
Live registry read from the gateway's own API — this is the actual allowlist the agent runs against, not an illustration. Every tool call passes through here; what is not on the allowlist cannot be called.

gateway healthy

ContextForge (mcp-contextforge-gateway) · virtual server "change-approval"

[ContextForge Admin UI ↗](#)

Agent's actual scope



Registered on the gateway but NOT in the agent's scope:

🚫 restart_service (ops-server)

Existing is not the same as reachable — calls to these from the agent are denied and audited (refusal ☹).

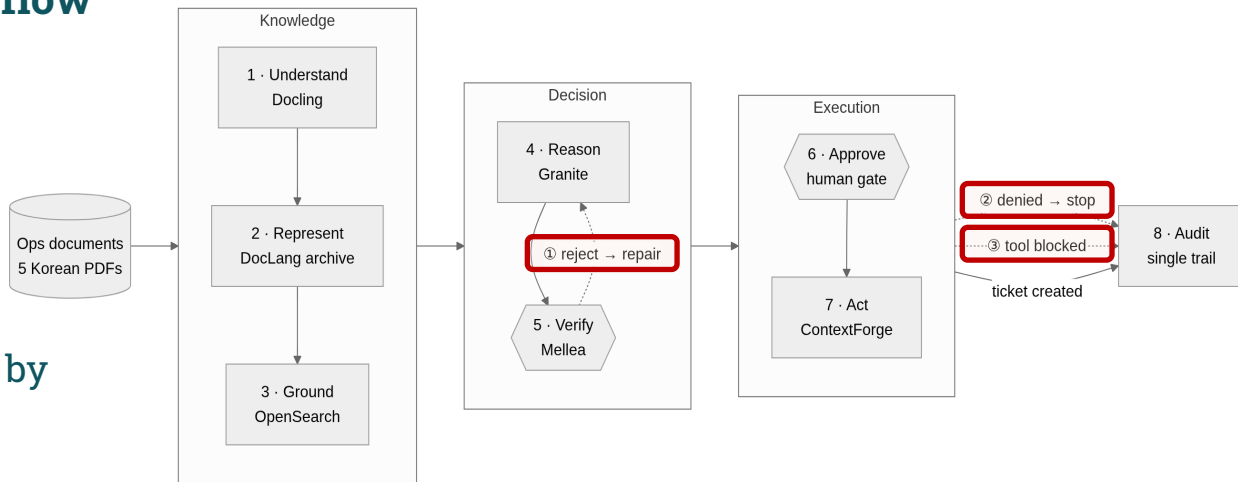
Registered MCP servers

SERVER	ENDPOINT	TRANSPORT	TOOL	ON ALLOWLIST?
ops-server	http://localhost:8101/mcp	STREAMABLEHTTP	restart_service	excluded
ticket-server	http://localhost:8100/mcp	STREAMABLEHTTP	create_change_ticket	allowed

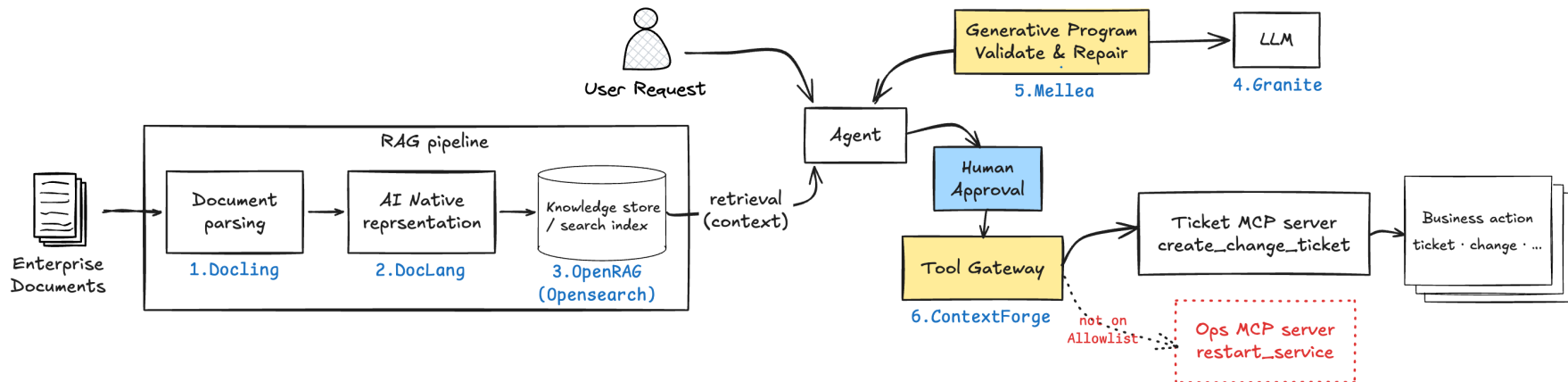
Trust Mechanism on Workflow Chain

The Three Refusals on Workflow

1. Verify Layer by **Mellea**
2. Approval **Human Gate**
3. Tool Audit and Agent Control by **ContextForge**



IBM Open Source Generative AI Ecosystem



¹ Docling

Understand documents



² DocLang

Preserve document structure



³ OpenRAG

Retrieve grounded knowledge



⁴ Granite

Reason & Generate



⁵ Mellea

Validate & Repair outputs



⁶ ContextForge

Govern tool access & execution



Session Takeaways

- ✓ Reliability is an architecture problem, not a model-only problem
- ✓ Every workflow stage requires dedicated trust mechanisms
- ✓ Open-source patterns can provide scalable governance and control
- ✓ Enterprise AI must be observable, auditable, and predictableReliability must be engineered across the stack

Goal: Build AI systems that enterprises can trust in production.





OPEN
KOREA

SOURCE SUMMIT

THE LINUX FOUNDATION

서울

