```
# OpenPipeline Migration Guide: Part 2

> **Series:** OPMIG | **Notebook:** 2 of 9 | **Created:** December 2025

## Architecture & Key Concepts

---

## Learning Objectives

By the end of this notebook, you will:

- ✅ Understand the complete OpenPipeline data flow architecture
- ✅ Learn detailed processing stage execution order
- ✅ Master DPL (Dynatrace Pattern Language) fundamentals
- ✅ **Know ALL OpenPipeline limits in comprehensive detail**
- ✅ Understand processor types and capabilities
- ✅ **Learn entity field availability timeline**
- ✅ Explore your pipeline configuration with DQL

---

---

## Data Flow Architecture

Understanding how data flows through OpenPipeline is essential for designing
effective pipelines.

### Complete Data Flow

![OpenPipeline Architecture]
(
ciIHZpZXdCb3g9IjAgMCA4MDAgMzIwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
iaW5nZXN0R3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+CiAgICAgIDxz
dG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMzYjgyZjY7c3RvcC1vcGFjaXR5OjEiI
C8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzI1NjNlYjtzdG
9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGllbnQ
gaWQ9InJvdXRlR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+CiAgICAg
IDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM4YjVjZjY7c3RvcC1vcGFjaXR5O
jEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzdjM2FlZD
tzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGl
lbnQgaWQ9InByb2Nlc3NHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4K
ICAgICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2Y1OWUwYjtzdG9wLW9wY
WNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZD
k3NzA2O3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgPC9kZWZzPgog
HcmFkaWVudCBpZD0iZXh0cmFjdEdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEw
```

MCUiPgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMTBiOTgxO3N0b
3Atb3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG
9yOiMwNTk2Njk7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGx
pbmVhckdyYWRpZW50IGlkPSJzdG9yZUdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9
IjEwMCUiPgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZWY0NDQ0O
3N0b3Atb3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLW
NvbG9yOiNkYzI2MjY7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICA
gPGZpbHRlciBpZD0ib3BTaGFkb3ciPgogICAgICA8ZmVEcm9wU2hhZG93IGR4PSIyIiBkeT0iMiIg
c3RkRGV2aWF0aW9uPSIzIiBmbG9vZC1vcGFjaXR5PSIwLjIiLz4KICAgIDwvZmlsdGVyPgogICAgP
G1hcmtlciBpZD0ib3BBcnJvdyIgbWFya2VyV2lkdGg9IjEwIiBtYXJrZXJIZWlnaHQ9IjciIHJlZl
g9IjkiIHJlZlk9IjMuNSIgb3JpZW50PSJhdXRvIj4KICAgICAgPHBvbHlnb24gcG9pbnRzPSIwIDA
sIDEwIDMuNSwgMCA3IiBmaWxsPSIjNjQ3NDhiIi8+CiAgICA8L21hcmtlcj4KICA8L2RlZnM+Cgog
IDwhLS0gQmFja2dyb3VuZCAtLT4KICA8cmVjdCB3aWR0aD0iODAwIiBoZWlnaHQ9IjMyMCIgZmlsb
D0iI2Y4ZjlmYSIgcng9IjEwIi8+CgogIDwhLS0gVGl0bGUgLS0+CiAgPHRleHQgeD0iNDAwIiB5PS
IyOCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE4IiBmb250LXd
laWdodD0iYm9sZCIgZmlsbD0iIzMzMyIgdGV4dC1hbmNob3I9Im1pZGRsZSI+T3BlblBpcGVsaW5l
IEFyY2hpdGVjdHVyZTwvdGV4dD4KCiAgPCEtLSBTdGFnZSAxOiBJbmdlc3QgLS0+CiAgPHJlY3Qge
D0iMzAiIHk9IjYwIiB3aWR0aD0iMTIwIiBoZWlnaHQ9IjEwMCIgcng9IjgiIGZpbGw9InVybCgjaW
5nZXN0R3JhZCkiIGZpbHRlcj0idXJsKCNvcFNoYWRvdykiLz4KICA8dGV4dCB4PSI5MCIgeT0iOTA
iIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiIgZm9udC13ZWln
aHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5JTkdFU1Q8L3RleHQ+C
iAgPHRleHQgeD0iOTAiIHk9IjExMCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250
LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWR
kbGUiPk9uZUFnZW50PC90ZXh0PgogIDx0ZXh0IHg9IjkwIiB5PSIxMjQiIGZvbnQtZmFtaWx5PSJB
cmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwL
jkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Mb2cgQVBJPC90ZXh0PgogIDx0ZXh0IHg9IjkwIiB5PS
IxMzgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0
icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5PVExQPC90ZXh0Pgog
IDx0ZXh0IHg9IjkwIiB5PSIxNTIiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9ud
C1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZG
xlIj5HZW5lcmljIEFQSTwvdGV4dD4KCiAgPCEtLSBBcnJvdyAxIC0tPgogIDxwYXRoIGQ9Ik0xNTA
sMTEwIEwxNzUsMTEwIiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9u
ZSIgbWFya2VyLWVuZD0idXJsKCNvcEFycm93KSIvPgoKICA8IS0tIFN0YWdlIDI6IFJvdXRpbmcgL
S0+CiAgPHJlY3QgeD0iMTg1IiB5PSI2MCIgd2lkdGg9IjEyMCIgaGVpZ2h0PSIxMDAiIHJ4PSI4Ii
BmaWxsPSJ1cmwoI3JvdXRlR3JhZCkiIGZpbHRlcj0idXJsKCNvcFNoYWRvdykiLz4KICA8dGV4dCB
4PSIyNDUiIHk9IjkwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0i
MTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U
k9VVElORzwvdGV4dD4KICA8dGV4dCB4PSIyNDUiIHk9IjExMCIgZm9udC1mYW1pbHk9IkFyaWFsLC
BzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHR
leHQtYW5jaG9yPSJtaWRkbGUiPlBlbGluZTwvdGV4dD4KICA8dGV4dCB4PSIyNDUiIHk9IjEyNCIg
IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnY
mEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U2VsZWN0aW9uPC90ZXh0Pg
ogIDx0ZXh0IHg9IjI0NSIgeT0iMTM4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZ
vbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1p
ZGRsZSI+TWF0Y2hpbmc8L3RleHQ+CiAgPHRleHQgeD0iMjQ1IiB5PSIxNTIiIGZvbnQtZmFtaWx5P
SJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NS
wwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Db25kaXRpb25zPC90ZXh0PgoKICA8IS0tIEFycm9
3IDIgLS0+CiAgPHBhdGggZD0iTTMwNSwxMTAgTDMzMCwxMTAiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ry

```
b2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI29wQXJyb3cpIi8+CgogI
DwhLS0gU3RhZ2UgMzogUHJvY2Vzc2luZyAtLT4KICA8cmVjdCB4PSIzNDAiIHk9IjYwIiB3aWR0aD
0iMTIwIiBoZWlnaHQ9IjEwMCIgcng9IjgiIGZpbGw9InVybCgjcHJvY2Vzc0dyYWQpIiBmaWx0ZXI
9InVybCgjb3BTaGFkb3cpIi8+CiAgPHRleHQgeD0iNDAwIiB5PSI5MCIgZm9udC1mYW1pbHk9IkFy
aWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id
2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlBST0NFU1M8L3RleHQ+CiAgPHRleHQgeD0iNDAwIi
B5PSIxMTAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZml
sbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5NYXNraW5nPC90
ZXh0PgogIDx0ZXh0IHg9IjQwMCIgeT0iMTI0IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2Vya
WYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3
I9Im1pZGRsZSI+UGFyc2luZzwvdGV4dD4KICA8dGV4dCB4PSI0MDAiIHk9IjEzOCIgZm9udC1mYW1
pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUs
MjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlRyYW5zZm9ybTwvdGV4dD4KICA8dGV4dCB4P
SI0MDAiIHk9IjE1MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9Ij
EwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkRyb3A
8L3RleHQ+CgogIDwhLS0gQXJyb3cgMyAtLT4KICA8cGF0aCBkPSJNNDYwLDExMCBMNDg1LDExMCIg
c3Ryb2tlPSIjNjQ3NDhiIiBzdHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmUiIG1hcmtlci1lbmQ9I
nVybCgjb3BBcnJvdykiLz4KCiAgPCEtLSBTdGFnZSA0OiBFeHRyYWN0IC0tPgogIDxyZWN0IHg9Ij
Q5NSIgeT0iNjAiIHdpZHRoPSIxMjAiIGhlaWdodD0iMTAwIiByeD0iOCIgZmlsbD0idXJsKCNleHR
yYWN0R3JhZCkiIGZpbHRlcj0idXJsKCNvcFNoYWRvdykiLz4KICA8dGV4dCB4PSI1NTUiIHk9Ijkw
IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ
2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RVhUUkFDVDwvdGV4dD
4KICA8dGV4dCB4PSI1NTUiIHk9IjExMCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiB
mb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJt
aWRkbGUiPk1ldHJpY3M8L3RleHQ+CiAgPHRleHQgeD0iNTU1IiB5PSIxMjQiIGZvbnQtZmFtaWx5P
SJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NS
wwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5FdmVudHM8L3RleHQ+CiAgPHRleHQgeD0iNTU1IiB
5PSIxMzgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmls
bD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5CaXpldmVudHM8L
3RleHQ+CiAgPHRleHQgeD0iNTU1IiB5PSIxNTIiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZX
JpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2h
vcj0ibWlkZGxlIj5BdHRyaWJ1dGVzPC90ZXh0PgoKICA8IS0tIEFycm93IDQgLS0+CiAgPHBhdGgg
ZD0iTTYxNSwxMTAgTDY0MCwxMTAiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ryb2tlLXdpZHRoPSIyIiBma
WxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI29wQXJyb3cpIi8+CgogIDwhLS0gU3RhZ2UgNTogU3
RvcmUgLS0+CiAgPHJlY3QgeD0iNjUwIiB5PSI2MCIgd2lkdGg9IjEyMCIgaGVpZ2h0PSIxMDAiIHJ
4PSI4IiBmaWxsPSJ1cmwoI3N0b3JlR3JhZCkiIGZpbHRlcj0idXJsKCNvcFNoYWRvdykiLz4KICA8
dGV4dCB4PSI3MTAiIHk9IjkwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc
2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZG
RsZSI+U1RPUkU8L3RleHQ+CiAgPHRleHQgeD0iNzEwIiB5PSIxMTAiIGZvbnQtZmFtaWx5PSJBcml
hbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkp
IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5HcmFpbDwvdGV4dD4KICA8dGV4dCB4PSI3MTAiIHk9IjEyN
CIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2
JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkJ1Y2tldHM8L3RleHQ+CiA
gPHRleHQgeD0iNzEwIiB5PSIxMzgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9u
dC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZ
GxlIj5SZXRlbnRpb248L3RleHQ+CiAgPHRleHQgeD0iNzEwIiB5PSIxNTIiIGZvbnQtZmFtaWx5PS
JBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSw
wLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Sb3V0aW5nPC90ZXh0PgoKICA8IS0tIEtleSBGZWF0
```

dXJlcyBTZWN0aW9uIC0tPgogIDxyZWN0IHg9IjMwIiB5PSIxODAiIHdpZHRoPSI3NDAiIGhlaWdod
D0iMTI1IiByeD0iOCIgZmlsbD0iI2ZmZiIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva2Utd2lkdGg9Ij
IiLz4KICA8dGV4dCB4PSI0MDAiIHk9IjIwNSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcml
mIiBmb250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzMzMyIgdGV4dC1hbmNo
b3I9Im1pZGRsZSI+S2V5IFByaW5jaXBsZXM8L3RleHQ+CgogIDwhLS0gRmVhdHVyZSBib3hlcyAtL
T4KICA8cmVjdCB4PSI1MCIgeT0iMjIwIiB3aWR0aD0iMTYwIiBoZWlnaHQ9IjcwIiByeD0iNiIgZm
lsbD0iI2RiZWFmZSIvPgogIDx0ZXh0IHg9IjEzMCIgeT0iMjQ1IiBmb250LWZhbWlseT0iQXJpYWws
sIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMWU0
MGFmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5QcmUtU3RvcmFnZTwvdGV4dD4KICA8dGV4dCB4PSIxM
zAiIHk9IjI2MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIi
BmaWxsPSIjMWU0MGFmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5CbGwgcHJvY2Vzc2luZyBiZWZvcmU
8L3RleHQ+CiAgPHRleHQgeD0iMTMwIiB5PSIyNzYiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1z
ZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzFlNDBhZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+Z
GF0YSBpcyBwZXJzaXN0ZWQ8L3RleHQ+CgogIDxyZWN0IHg9IjIzMCIgeT0iMTYwIiB3aWR0aD0iMT
YwIiBoZWlnaHQ9IjcwIiByeD0iNiIgZmlsbD0iI2ZlZjNjNyIvPgogIDx0ZXh0IHg9IjMxMCIgeT0
iMjQ1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQt
d2VpZ2h0PSJib2xkIiBmaWxsPSIjOTI0MDBlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5NdWx0aS1Qa
XBlbGluZTwvdGV4dD4KICA8dGV4dCB4PSIzMTAiIHk9IjI2MiIgZm9udC1mYW1pbHk9IkFyaWFsLC
BzYW5zLXNlcmlmIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iOTI
kZGxlIj5VcCB0byA1IHBpcGVsaW5lcyBwZXI8L3RleHQ+CiAgPHRleHQgeD0iMzEwIiB5PSIyNzYi
IGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzkyN
DAwZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+cmVjb3JkIHNpbXVsdGFuZW91c2x5PC90ZXh0PgoKIC
A8cmVjdCB4PSI0MTAiIHk9IjIyMCIgd2lkdGg9IjE2MCIgaGVpZ2h0PSI3MCIgcng9IjYiIGZpbGw
9IiNkMWZhZTUiLz4KICA8dGV4dCB4PSI0OTAiIHk9IjI0NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBz
YW5zLXNlcmlmIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzA0Nzg1N
yIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U2VjdXJpdHkgRmlyc3Q8L3RleHQ+CiAgPHRleHQgeD0iND
kwIiB5PSIyNjIiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCI
gZmlsbD0iIzA0Nzg1NyIgdGV4dC1hbmNob3I9Im1pZGRsZSI+TWFza2luZyBiZWZvcmUgYW55PC90
ZXh0PgogIDx0ZXh0IHg9IjQ5MCIgeT0iMjc2IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2Vya
WYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiMwNDc4NTciIHRleHQtYW5jaG9yPSJtaWRkbGUiPm90aG
VyIHByb2Nlc3Npbmc8L3RleHQ+CgogIDxyZWN0IHg9IjU5MCIgeT0iMTYwIiB3aWR0aD0iMTYwIiB
oZWlnaHQ9IjcwIiByeD0iNiIgZmlsbD0iI2ZjZTdmMyIvPgogIDx0ZXh0IHg9IjY3MCIgeT0iMjQ1
IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ
2h0PSJib2xkIiBmaWxsPSIjOWQxNzRkIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5FbnRpdHkgRGV0ZW
N0aW9uPC90ZXh0PgogIDx0ZXh0IHg9IjY3MCIgeT0iMjYyIiBmb250LWZhbWlseT0iQXJpYWwsIHNhb
hbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM5ZDE3NGQiIHRleHQtYW5jaG9yPSJtaWRk
bGUiPmR0LmVudGl0eS4qIGFkZGVkIGFmdGVyPC90ZXh0PgogIDx0ZXh0IHg9IjY3MCIgeT0iMjc2I
iBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM5ZD
E3NGQiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlByb2Nlc3Npbmcgc3RhZ2U8L3RleHQ+Cjwvc3ZnPgo
=)

Key Principles

1. **Pre-storage Processing**: All transformations happen BEFORE data is written to Grail
2. **Order Matters**: Processors execute in defined order within each stage

````
3. **Multi-pipeline Support**: A single record can be processed by up to 5
pipelines
4. **Entity Detection**: Entity fields (`dt.entity.*`) are added AFTER the
Processing stage
5. **Immutable Storage**: Once stored, data cannot be modified (mask early!)

---

## Processing Stages in Detail

### Stage 1: Routing

The routing stage determines which pipeline(s) process each incoming record.

| Aspect | Description |
|--------|-------------|
| **Purpose** | Match incoming data to appropriate pipelines |
| **Timing** | First stage - before any processing |
| **Configuration** | Dynamic routing rules with matching conditions |
| **Fallback** | Unmatched data goes to default pipeline |

**Matching Condition Examples:**
```
k8s.namespace.name == "production"
log.source == "nginx"
contains(content, "payment")
dt.openpipeline.source == "oneagent"
```

### Stage 2: Processing

The processing stage transforms, enriches, and filters data.

**Sub-stages (in order):**

1. **Masking** - Redact sensitive data (applied FIRST for security)
2. **Filtering** - Drop unwanted records
3. **Processing** - Parse, transform, enrich

| Processor Type | Purpose | Example |
|----------------|---------|----------|
| **DQL** | Transform with DQL commands | `fieldsAdd`, `fieldsRemove`,
`parse` |
| **Drop** | Remove matching records | Drop debug logs |
| **Technology Parser** | Apply built-in parsers | Apache, JSON, syslog |

### Stage 3: Extraction
````

```
The extraction stage creates derived data from processed records.

| Extraction Type | Output | Use Case |
|-----------------|--------|----------|
| **Value Metric** | Numeric metric with dimensions | Response times, counts
|
| **Counter Metric** | Incrementing counter | Request counts, errors |
| **Event** | Platform event | Custom alerts, notifications |
| **Business Event** | Business analytics event | Transactions, user actions
|

### Stage 4: Storage

The storage stage routes data to Grail buckets.

| Aspect | Description |
|--------|-------------|
| **Bucket Routing** | Direct data to specific buckets |
| **Retention** | Different retention periods per bucket |
| **Cost Control** | Route high-volume data to shorter retention |

---

## Pipeline Types

OpenPipeline supports three types of pipelines:

### 1. Default Pipeline

- **Purpose**: Handles data that doesn't match any custom pipeline
- **Configuration**: Minimal processing, default bucket
- **Modification**: Can add processors but cannot delete

### 2. Custom Pipelines

- **Purpose**: Targeted processing for specific data sources or patterns
- **Configuration**: Full control over all processing stages
- **Routing**: Requires matching conditions in dynamic routing

### 3. Built-in Pipelines

- **Purpose**: Pre-configured for common technologies
- **Examples**: Kubernetes logs, AWS logs, Azure logs
- **Modification**: Can extend but not fundamentally change

> 💡 **Best Practice:** Create focused pipelines for specific use cases
rather than one large pipeline with complex conditionals.
```

````
---

## Processor Types

### DQL Processor

The DQL processor uses DQL commands to transform data. Available commands:

| Command | Purpose | Example |
|---------|---------|----------|
| `fieldsAdd` | Add new fields | `fieldsAdd environment = "prod"` |
| `fieldsRemove` | Remove fields | `fieldsRemove sensitive_field` |
| `fieldsRename` | Rename fields | `fieldsRename old_name = new_name` |
| `parse` | Extract with DPL patterns | `parse content, "LD:prefix
INT:count"` |

**fieldsAdd Examples:**
```dql
// Static value
| fieldsAdd environment = "production"

// Conditional value
| fieldsAdd severity = if(loglevel == "ERROR", "critical", else: "normal")

// Computed value
| fieldsAdd message_length = stringLength(content)

// From existing field
| fieldsAdd short_host = substring(host.name, 0, 10)
```

### Drop Processor

Removes records matching specified conditions:

```
Matching Condition: loglevel == "DEBUG"
Result: All DEBUG logs are dropped before storage
```

### Technology Processors

Built-in parsers for common log formats:

| Technology | Parsed Fields |
|------------|---------------|
| Apache | client_ip, method, path, status, bytes |
| Nginx | Similar to Apache with nginx-specific fields |
````

````
| JSON | Flattens JSON structure to fields |
| Syslog | facility, severity, hostname, message |

---

## Dynatrace Pattern Language (DPL)

DPL is a powerful pattern matching language used in the `parse` command.

### Core Matchers

| Matcher | Description | Example Match |
|---------|-------------|---------------|
| `INT` | Integer number | `42`, `-17` |
| `LONG` | Long integer | `1234567890123` |
| `DOUBLE` | Decimal number | `3.14`, `-0.5` |
| `IPADDR` | IPv4 or IPv6 address | `192.168.1.1`, `::1` |
| `IPV4ADDR` | IPv4 address only | `10.0.0.1` |
| `IPV6ADDR` | IPv6 address only | `2001:db8::1` |
| `TIMESTAMP` | Timestamp with format | `2024-01-15T10:30:00Z` |
| `LD` | Line data (to delimiter) | Any text until delimiter |
| `DATA` | Any data (greedy) | Consumes remaining text |
| `SPACE` | Whitespace | Spaces, tabs |
| `NSPACE` | Non-whitespace | Any non-space chars |
| `WORD` | Word characters | `hello`, `user123` |
| `JSON` | JSON structure | `{"key": "value"}` |
| `EOL` | End of line | Line terminator |

### Pattern Syntax

| Element | Syntax | Description |
|---------|--------|-------------|
| Export to field | `MATCHER:fieldname` | Extract and name the field |
| Match only | `MATCHER` | Match but don't extract |
| Optional | `MATCHER?` | Matcher is optional |
| Literal | `'exact text'` | Match literal string |
| Alternatives | `('opt1'\|'opt2')` | Match either option |
| Quantifier | `MATCHER{2,5}` | Match 2-5 times |

### DPL Examples

```dql
// Extract IP and port from log
| parse content, "IPADDR:client_ip ':' INT:port"

// Extract user ID with flexible prefix
| parse content, "('user='|'userId='|'user_id=')LD:user_id"
````
````

```
// Parse Apache-style log
| parse content, "IPADDR:client_ip SPACE '-' SPACE LD:user SPACE '['
LD:timestamp ']'"

// Extract JSON payload
| parse content, "LD JSON:payload"

// Parse with optional port
| parse content, "IPADDR:ip (':' INT:port)?"

// Extract error code
| parse content, "'error_code=' INT:error_code"
```

Timestamp Format Patterns

Symbol	Meaning	Example
`yyyy`	Year (4 digits)	2024
`MM`	Month (01-12)	01
`dd`	Day (01-31)	15
`HH`	Hour 24h (00-23)	14
`mm`	Minute (00-59)	30
`ss`	Second (00-59)	45
`SSS`	Milliseconds	123

Complete OpenPipeline Limits Reference

This comprehensive reference contains ALL OpenPipeline limits you need to know for planning your migration.

Data Size & Volume Limits

Limit	Value	Behavior When Exceeded
Max record size (after processing)	16 MB	Record is **dropped**, logged in audit
Max record size (before processing)	1 MB	Ingestion rejected with 413 error
Working memory per record	16 MB	Processing fails, record dropped
Log attribute size	32 KB	Attribute value **truncated**
Max field name length	255 characters	Field creation fails
Max string field length	4 KB	Content truncated
Max array size	1000 elements	Array truncated
Max nesting depth (JSON)	10 levels	Deeper levels flattened

```
### Processing Limits

| Limit | Value | Impact |
|-------|-------|--------|
| **Max pipelines per record** | 5 | Up to 5 pipelines can process one record
|
| **Max processors per pipeline** | 50 | Cannot add more processors to
pipeline |
| **Max DQL commands per processor** | 10 commands | Split complex logic into
multiple processors |
| **Max parse operations per processor** | 100 patterns | Create additional
parse processors |
| **Max fields per record** | 1000 fields | Additional fields ignored |
| **Processing timeout per record** | 30 seconds | Record dropped if exceeded
|
| **Max processor name length** | 100 characters | Validation error |

### Timestamp Constraints

| Data Type | Accepted Range | Records Outside Range |
|-----------|---------------|----------------------|
| **Logs** | 24 hours past to 10 minutes future | **Dropped** |
| **Spans** | 2 hours past | **Dropped** |
| **Events** | 24 hours past to 10 minutes future | **Dropped** |
| **Business Events** | 24 hours past to 10 minutes future | **Dropped** |
| **Metrics** | 1 hour past to 1 minute future | **Dropped** |

> ⚠️ **Critical:** Historical data imports require workarounds. Contact
Dynatrace support for backfilling options.

### Pipeline & Routing Limits

| Limit | Value | Scope |
|-------|-------|-------|
| **Max custom pipelines** | 100 pipelines | Per configuration scope (logs,
spans, etc.) |
| **Max dynamic routes** | 100 routes | Per configuration scope |
| **Max conditions per route** | 10 conditions | Combine with AND/OR
operators |
| **Max pipeline name length** | 100 characters | Validation error |
| **Max route name length** | 100 characters | Validation error |

### Extraction Limits

| Limit | Value | Notes |
|-------|-------|-------|
| **Max metric extractions per pipeline** | 10 | Value + counter metrics
combined |
```

````
| **Max event extractions per pipeline** | 5 | All event types combined |
| **Max bizevent extractions per pipeline** | 3 | Business events only |
| **Max dimensions per metric** | 10 dimensions | Keep cardinality low |
| **Metric key length** | 250 characters | Validation error |
| **Event type name length** | 100 characters | Validation error |

### DQL & DPL Limits

| Limit | Value | Context |
|-------|-------|------|
| **Max DQL query length** | 64 KB | In processor definition |
| **Max DPL pattern length** | 4 KB | Per parse pattern |
| **Max captured groups per parse** | 100 groups | Use multiple parse
operations |
| **Max alternatives in pattern** | 50 alternatives |
`(opt1\|opt2\|...\|opt50)` |

### Bucket & Storage Limits

| Limit | Value | Notes |
|-------|-------|-------|
| **Max custom buckets** | 50 buckets | Per environment |
| **Min retention period** | 1 day | Per bucket |
| **Max retention period** | 2555 days (~7 years) | Per bucket |
| **Bucket name length** | 100 characters | Alphanumeric + underscore |

### Rate Limits

| Endpoint | Limit | Per |
|----------|-------|-----|
| **Log Ingest API** | 500 requests/min | Per token |
| **OTLP Endpoint** | 1000 requests/min | Per token |
| **Config API** | 100 requests/min | Per token |

### Field & Matching Restrictions

**Read-Only Fields** (Cannot be modified):
```
dt.ingest.*          - Ingestion metadata
dt.openpipeline.*    - Pipeline processing metadata
dt.retain.*          - Retention information
dt.system.*          - System metadata (bucket, etc.)
timestamp            - Record timestamp (can parse new, not modify original)
```

**Reserved Field Prefixes** (Cannot create):
```
dt.*                 - Reserved for Dynatrace
````
````

````
dynatrace.*          - Reserved for Dynatrace
```

**Entity Fields** (Available ONLY after Processing stage):
```
dt.entity.service
dt.entity.host
dt.entity.process_group
dt.entity.process_group_instance
dt.entity.kubernetes_cluster
dt.entity.cloud_application
dt.entity.cloud_application_namespace
```

> 💡 **Design Pattern:** Entity fields are added by Dynatrace AFTER the Processing stage. You cannot:
> - Use them in routing conditions
> - Use them in matching conditions during processing
> - Modify them with fieldsAdd/fieldsRename
>
> You CAN use them in:
> - Extraction stage (metrics, events)
> - Storage stage (bucket routing based on entity)

### Workarounds for Common Limit Issues

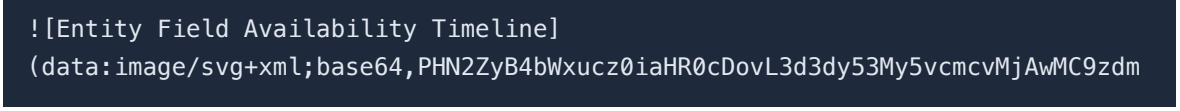
| Problem | Workaround |
|---------|------------|
| **>50 processors needed** | Split into multiple pipelines, use multi-pipeline routing |
| **>10 DQL commands** | Break into multiple processors (order matters!) |
| **>100 parse patterns** | Use multiple parse processors in sequence |
| **>16MB after processing** | Drop unnecessary fields, reduce field sizes |
| **>5 pipelines needed** | Consolidate processing logic, use conditional processors |
| **>10 metric dimensions** | Reduce cardinality, use separate metrics |

---

## Entity Field Availability Timeline

Understanding WHEN entity fields become available is critical for pipeline design.

### Processing Stage Timeline



````
````

```
ciIHZpZXdCb3g9IjAgMCA3MDAgNDUwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
iaW5nZXN0R3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+CiAgICAgIDxz
dG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM2NDc0OGI7c3RvcC1vcGFjaXR5OjEiI
C8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzQ3NTU2OTtzdG
9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGllbnQ
gaWQ9InJvdXRlR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+CiAgICAg
IDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM2NDc0OGI7c3RvcC1vcGFjaXR5O
jEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzQ3NTU2OT
tzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGl
lbnQgaWQ9InByb2Nlc3NHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4K
ICAgICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzY0NzQ4YjtzdG9wLW9wY
WNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojND
c1NTY5O3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJ
HcmFkaWVudCBpZD0iZGV0ZWN0R3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAw
JSI+CiAgICAgIDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM4YjVjZjY7c3Rvc
C1vcGFjaXR5OjEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3
I6IzdjM2FlZDtzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGl
uZWFyR3JhZGllbnQgaWQ9ImV4dHJhY3RHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHky
PSIxMDAlIj4KICAgICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzEwYjk4M
TtzdG9wLW9wYWNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC
1jb2xvcjojMDU5NjY5O3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICA
gIDxsaW5lYXJHcmFkaWVudCBpZD0ic3RvcmVHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUi
IHkyPSIxMDAlIj4KICAgICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzEwY
jk4MTtzdG9wLW9wYWNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3
RvcC1jb2xvcjojMDU5NjY5O3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4
KICAgIDxmaWx0ZXIgaWQ9ImVmU2hhZG93Ij4KICAgICAgPGZlRHJvcFNoYWRvdyBkeD0iMSIgZHk9
IjEiIHN0ZERldmlhdGlvbj0iMiIgZmxvb2Qtb3BhY2l0eT0iMC4xNSIvPgogICAgPC9maWx0ZXI+C
iAgICA8bWFya2VyIGlkPSJlZkFycm93IiBtYXJrZXJXaWR0aD0iMTAiIG1hcmtlckhlaWdodD0iNy
IgcmVmWD0iOSIgcmVmWT0iMy41IiBvcmllbnQ9ImF1dG8iPgogICAgICA8cG9seWdvbiBwb2ludHM
9IjAgMCwgMTAgMy41LCAwIDciIGZpbGw9IiM2NDc0OGIiLz4KICAgIDwvbWFya2VyPgogIDwvZGVm
cz4KCiAgPCEtLSBCYWNrZ3JvdW5kIC0tPgogIDxyZWN0IHdpZHRoPSI3MDAiIGhlaWdodD0iNDUwI
iBmaWxsPSIjZjhmOWZhIiByeD0iMTAiLz4KCiAgPCEtLSBUaXRsZSAtLT4KICA8dGV4dCB4PSIzNT
AiIHk9IjI4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTgiIGZ
vbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5FbnRpdHkg
RmllbGQgQXZhaWxhYmlsaXR5IFRpbWVsaW5lPC90ZXh0PgogIDx0ZXh0IHg9IjM1MCIgeT0iNDgiI
GZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiIgZmlsbD0iIzY2Ni
IgdGV4dC1hbmNob3I9Im1pZGRsZSI+ZHQuZW50aXR5LiogZmllbGRzIGFyZSBhZGRlZCBBRlUiB
0aGUgUHJvY2Vzc2luZyBzdGFnZTwvdGV4dD4KCiAgPCEtLSBUaW1lbGluZSBsaW5lIC0tPgogIDxs
aW5lIHgxPSIxMDAiIHkxPSI5MCIgeDI9IjYwMCIgeTI9IjkwIiBzdHJva2U9IiNjYmQ1ZTEiIHN0c
m9rZS13aWR0aD0iMyIvPgoKICA8IS0tIFN0YWdlIDE6IEluZ2VzdCAtLT4KICA8Y2lyY2xlIGN4PS
IxMjAiIGN5PSI5MCIgcj0iMTgiIGZpbGw9InVybCgjaW5nZXN0R3JhZCkiIGZpbHRlcj0idXJsKCN
lZlNoYWRvdykiLz4KICA8dGV4dCB4PSIxMjAiIHk9Ijk1IiBmb250LWZhbWlseT0iQXJpYWwsIHNh
bnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgd
GV4dC1hbmNob3I9Im1pZGRsZSI+MTwvdGV4dD4KICA8dGV4dCB4PSIxMjAiIHk9IjEyNSIgZm9udC
1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9
sZCIgZmlsbD0iIzQ3NTU2OSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+SU5HRVNUPC90ZXh0PgogIDxy
ZWN0IHg9Ijc1IiB5PSIxMzUiIHdpZHRoPSI5MCIgaGVpZ2h0PSIyNCIgcng9IjEyIiBmaWxsPSIjZ
mVjYWNhIi8+CiAgPHRleHQgeD0iMTIwIiB5PSIxNTEiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy
```

1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2RjMjYyNiIgdGV4dC1hbmNob3I9Im1pZGRsZSI
+4p2MIE5vIEVudGl0eTwvdGV4dD4KCiAgPCEtLSBTdGFnZSAyOiBSb3V0aW5nIC0tPgogIDxjaXJj
bGUgY3g9IjIyMCIgY3k9IjkwIiByPSIxOCIgZmlsbD0idXJsKCNyb3V0ZUdyYWQpIiBmaWx0ZXI9I
nVybCgjZWZTaGFkb3cpIi8+CiAgPHRleHQgeD0iMjIwIiB5PSI5NSIgZm9udC1mYW1pbHk9IkFyaW
FsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2h
pdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjI8L3RleHQ+CiAgPHRleHQgeD0iMjIwIiB5PSIxMjUi
IGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlna
HQ9ImJvbGQiIGZpbGw9IiM0NzU1NjkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlJPVVRJTkc8L3RleH
Q+CiAgPHJlY3QgeD0iMTc1IiB5PSIxMzUiIHdpZHRoPSI5MCIgaGVpZ2h0PSIyNCIgcng9IjEyIiB
maWxsPSIjZmVjYWNhIi8+CiAgPHRleHQgeD0iMjIwIiB5PSIxNTEiIGZvbnQtZmFtaWx5PSJBcmlh
bCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2RjMjYyNiIgdGV4dC1hbmNob3I9I
m1pZGRsZSI+4p2MIE5vIEVudGl0eTwvdGV4dD4KCiAgPCEtLSBTdGFnZSAzOiBQcm9jZXNzaW5nIC
0tPgogIDxjaXJjbGUgY3g9IjMyMCIgY3k9IjkwIiByPSIxOCIgZmlsbD0idXJsKCNwcm9jZXNzR3J
hZCkiIGZpbHRlcj0idXJsKCNlZlNoYWRvdykiLz4KICA8dGV4dCB4PSIzMjAiIHk9Ijk1IiBmb250
LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib
2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MzwvdGV4dD4KICA8dGV4dCB4PS
IzMjAiIHk9IjEyNSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE
xIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzQ3NTU2OSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+
UFJPQ0VTU0lORzwvdGV4dD4KICA8cmVjdCB4PSIyNzUiIHk9IjEzNSIgd2lkdGg9IjkwIiBoZWlna
HQ9IjI0IiByeD0iMTIiIGZpbGw9IiNmZWNhY2EiLz4KICA8dGV4dCB4PSIzMjAiIHk9IjE1MSIgZm
9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjZGMyNjI
2IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj7inYwgTm8gRW50aXR5PC90ZXh0PgoKICA8IS0tIFN0YWdl
IDQ6IEVudGl0eSBEZXRlY3Rpb24gKGhpZ2hsaWdodGVkKSAtLT4KICA8Y2lyY2xlIGN4PSI0MjAiI
GN5PSI5MCIgcj0iMjIiIGZpbGw9InVybCgjZGV0ZWN0R3JhZCkiIHN0cm9rZT0iI2M0YjVmZCIgc3
Ryb2tlLXdpZHRoPSIzIiBmaWx0ZXI9InVybCgjZWZTaGFkb3cpIi8+CiAgPHRleHQgeD0iNDIwIiB
5PSI5NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250
LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPuKtkDwvdGV4d
D4KICA8dGV4dCB4PSI0MjAiIHk9IjEyNSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIi
Bmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzdjM2FlZCIgdGV4dC1hbmN
ob3I9Im1pZGRsZSI+RU5USVRZPC90ZXh0PgogIDxyZWN0IHg9IjM2MCIgeT0iMTM1IiB3aWR0aD0i
MTIwIiBoZWlnaHQ9IjI0IiByeD0iMTIiIGZpbGw9IiNlZGU5ZmUiIHN0cm9rZT0iIzhiNWNmNiIgc
3Ryb2tlLXdpZHRoPSIyIi8+CiAgPHRleHQgeD0iNDIwIiB5PSIxNTEiIGZvbnQtZmFtaWx5PSJBcm
lhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzdjM2FlZCIgdGV4dC1hbmNob3I
9Im1pZGRsZSI+ZHQuZW50aXR5LiogQURERUQ8L3RleHQ+CgogIDwhLS0gU3RhZ2UgNTogRXh0cmFj
dGlvbiAtLT4KICA8Y2lyY2xlIGN4PSI1MjAiIGN5PSI5MCIgcj0iMTgiIGZpbGw9InVybCgjZXh0c
mFjdEdyYWQpIiBmaWx0ZXI9InVybCgjZWZTaGFkb3cpIi8+CiAgPHRleHQgeD0iNTIwIiB5PSI5NS
IgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaWd
odD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjU8L3RleHQ+CiAgPHRl
eHQgeD0iNTIwIiB5PSIxMjUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1za
XplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiMwNTk2NjkiIHRleHQtYW5jaG9yPSJtaW
RkbGUiPkVYVFJBQ1RJT048L3RleHQ+CiAgPHJlY3QgeD0iNDc1IiB5PSIxMzUiIHdpZHRoPSI5MCI
gaGVpZ2h0PSIyNCIgcng9IjEyIiBmaWxsPSIjZDFmYWU1Ii8+CiAgPHRleHQgeD0iNTIwIiB5PSIx
NTEiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI
zA1OTY2OSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4pyFIEF2YWlsYWJsZTwvdGV4dD4KCiAgPCEtLS
BTdGFnZSA2OiBTdG9yYWdlIC0tPgogIDxjaXJjbGUgY3g9IjYwMCIgY3k9IjkwIiByPSIxOCIgZml
sbD0idXJsKCNzdG9yZUdyYWQpIiBmaWx0ZXI9InVybCgjZWZTaGFkb3cpIi8+CiAgPHRleHQgeD0i
NjAwIiB5PSI5NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyI
iBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjY8L3

```
RleHQ+CiAgPHRleHQgeD0iNjAwIiB5PSIxMjUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJ
pZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiMwNTk2NjkiIHRleHQt
YW5jaG9yPSJtaWRkbGUiPlNUT1JBR0U8L3RleHQ+CiAgPHJlY3QgeD0iNTU1IiB5PSIxMzUiIHdpZ
HRoPSI5MCIgaGVpZ2h0PSIyNCIgcng9IjEyIiBmaWxsPSIjZDFmYWU1Ii8+CiAgPHRleHQgeD0iNj
AwIiB5PSIxNTEiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCI
gZmlsbD0iIzA1OTY2OSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4pyFIEF2YWlsYWJsZTwvdGV4dD4K
CiAgPCEtLSBEaXZpZGVyIC0tPgogIDxsaW5lIHgxPSI1MCIgeTE9IjE4NSIgeDI9IjY1MCIgeTI9I
jE4NSIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva2Utd2lkdGg9IjIiLz4KCiAgPCEtLSBXaGF0IFdvcm
tzIC8gV2hhdCBEb2Vzbid0IFNlY3Rpb24gLS0+CiAgPHRleHQgeD0iMzUwIiB5PSIyMTUiIGZvbnQ
tZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxNCIgZm9udC13ZWlnaHQ9ImJv
bGQiIGZpbGw9IiMzMzMiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkRlc2lnbiBJbXBsaWNhdGlvbnM8L
3RleHQ+CgogIDwhLS0gQ2FuJ3QgRG8gU2VjdGlvbiAtLT4KICA8cmVjdCB4PSI1MCIgeT0iMjM1Ii
B3aWR0aD0iMjkwIiBoZWlnaHQ9IjE5MCIgcng9IjgiIGZpbGw9IiNmZWYyZjIiIHN0cm9rZT0iI2Z
lY2FjYSIgc3Ryb2tlLXdpZHRoPSIyIi8+CiAgPHRleHQgeD0iMTk1IiB5PSIyNjAiIGZvbnQtZmFt
aWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiIgZm9udC13ZWlnaHQ9ImJvbGQiI
GZpbGw9IiNkYzI2MjYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPuKdjCBDQU5OT1QgRG8gKFN0YWdlcy
AxLTMpPC90ZXh0PgoKICA8dGV4dCB4PSI3MCIgeT0iMjkwIiBmb250LWZhbWlseT0iQXJpYWwsIHN
hbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiM3ZjFkMWQiPuKAoiBSb3V0ZSBiYXNlZCBv
biBkdC5lbnRpdHkuc2VydmljZTwvdGV4dD4KICA8dGV4dCB4PSI3MCIgeT0iMzE1IiBmb250LWZhb
WlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiM3ZjFkMWQiPuKAoi
BVc2UgZW50aXR5IGluIG1hdGNoaW5nIGNvbmRpdGlvbnM8L3RleHQ+CiAgPHRleHQgeD0iNzAiIHk
9IjM0MCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxs
PSIjN2YxZDFkIj7igKIgZmllbGRzQWRkIHVzaW5nIGR0LmVudGl0eS4qPC90ZXh0PgogIDx0ZXh0I
Hg9IjcwIiB5PSIzNjUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPS
IxMSIgZmlsbD0iIzdmMWQxZCI+4oCiIE1vZGlmeSBvciByZW5hbWUgZW50aXR5IGZpZWxkczwvdGV
4dD4KICA8dGV4dCB4PSI3MCIgeT0iMzkwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYi
IGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiM3ZjFkMWQiPuKAoiBEcm9wIHJlY29yZHMgYnkgZW50aXR5P
C90ZXh0PgoKICA8IS0tIENhbiBEbyBTZWN0aW9uIC0tPgogIDxyZWN0IHg9IjM2MCIgeT0iMjM1Ii
B3aWR0aD0iMjkwIiBoZWlnaHQ9IjE5MCIgcng9IjgiIGZpbGw9IiNmMGZkZjQiIHN0cm9rZT0iI2J
iZjdkMCIgc3Ryb2tlLXdpZHRoPSIyIi8+CiAgPHRleHQgeD0iNTA1IiB5PSIyNjAiIGZvbnQtZmFt
aWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiIgZm9udC13ZWlnaHQ9ImJvbGQiI
GZpbGw9IiMwNTk2NjkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPuKchSBDQU4gRG8gKFN0YWdlcyA1LT
YpPC90ZXh0PgoKICA8dGV4dCB4PSIzODAiIHk9IjI5MCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5
zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjMTY2NTM0Ij7igKIgVXNlIGVudGl0eSBhcyBt
ZXRyaWMgZGltZW5zaW9uPC90ZXh0PgogIDx0ZXh0IHg9IjM4MCIgeT0iMzE1IiBmb250LWZhbWlse
T0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiMxNjY1MzQiPuKAoiBJbm
NsdWRlIGVudGl0eSBpbiBldmVudCBkYXRhPC90ZXh0PgogIDx0ZXh0IHg9IjM4MCIgeT0iMzQwIiB
mb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiMxNjY1
MzQiPuKAoiBSb3V0ZSB0byBidWNrZXQgYnkgZW50aXR5PC90ZXh0PgogIDx0ZXh0IHg9IjM4MCIge
T0iMzY1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbG
w9IiMxNjY1MzQiPuKAoiBCdXNpbmVzcyBldmVudCB3aXRoIGVudGl0eSBjb250ZXh0PC90ZXh0Pgo
gIDx0ZXh0IHg9IjM4MCIgeT0iMzkwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZv
bnQtc2l6ZT0iMTEiIGZpbGw9IiMxNjY1MzQiPuKAoiBRdWVyeSBlbnRpdHkgZmllbGRzIGFmdGVyI
HN0b3JhZ2U8L3RleHQ+Cjwvc3ZnPgo=)
```


```
### Practical Examples
```

❌ WRONG - Cannot route based on entity
```
Dynamic Route:
  Condition: dt.entity.service == "SERVICE-123"  ❌ FAILS

Why: Entity fields not available during routing
```

❌ WRONG - Cannot use entity in processing
```dql
DQL Processor:
  fieldsAdd service_name = dt.entity.service  ❌ FAILS

Why: Entity fields not available during processing
```

✅ CORRECT - Use entity in metric extraction
```
Metric Extraction:
  Key: log.request.duration
  Value: duration_ms
  Dimensions: dt.entity.service, dt.entity.host  ✅ WORKS

Why: Entity fields available in extraction stage
```

✅ CORRECT - Use entity for bucket routing
```
Bucket Routing (in Storage stage):
  Condition: dt.entity.service == "CRITICAL-SERVICE"
  Bucket: critical_logs  ✅ WORKS

Why: Entity fields available in storage stage
```

Design Patterns

Pattern 1: Route by source, extract metrics by entity
```
Routing: log.source == "payment-service"  (uses source field)
Processing: Parse payment data
Extraction: Create metric with dt.entity.service dimension  ✅
```

Pattern 2: Enrich with custom field, then use entity
```
Processing: fieldsAdd app_tier = "frontend"  (custom field)
```

````
Extraction: Dimensions: app_tier, dt.entity.service  ✅
```

**Pattern 3: Cannot mix entity with routing**
```
❌ Route based on entity → Use workaround:
   - Route based on source/content patterns instead
   - Use service name string (not entity ID) if available
```

---

---

## Key Fields & Metadata

### OpenPipeline-Specific Fields

| Field | Description | Example Value |
|-------|-------------|---------------|
| `dt.openpipeline.source` | Data source identifier | `oneagent`, `generic`, `otlp` |
| `dt.openpipeline.pipelines` | Pipeline(s) that processed record | `["custom-pipeline-1"]` |
| `dt.system.bucket` | Grail storage bucket | `default_logs`, `custom_logs` |

### Log Fields

| Field | Description |
|-------|-------------|
| `timestamp` | Log timestamp (primary time field) |
| `content` | Log message content |
| `loglevel` | Log level (ERROR, WARN, INFO, DEBUG) |
| `status` | Status string (alternative to loglevel) |
| `log.source` | Log source identifier |
| `log.iostream` | Stream type (stdout, stderr) |

### Entity Context Fields

| Field | Description |
|-------|-------------|
| `dt.entity.host` | Host entity ID |
| `dt.entity.process_group` | Process group entity ID |
| `dt.entity.process_group_instance` | Process instance entity ID |
| `dt.entity.service` | Service entity ID |
| `host.name` | Host name (string) |
| `process.executable.name` | Process executable name |
````
````

````
> ⚠️ **Important:** Entity fields (`dt.entity.*`) are added AFTER the
Processing stage. You cannot use them in routing or processing conditions.

---

## Exploring Your Pipeline Configuration

Use these queries to understand how your environment is configured.

```python
// View data sources currently sending to OpenPipeline
// Shows the distribution of data by ingestion source
fetch logs, from: now() - 24h
| summarize {record_count = count()}, by: {dt.openpipeline.source}
| sort record_count desc
```

```python
// Analyze which pipelines are processing your logs
// Helps verify routing is working correctly
fetch logs, from: now() - 24h
| filter isNotNull(dt.openpipeline.pipelines)
| summarize {record_count = count()}, by: {dt.openpipeline.pipelines}
| sort record_count desc
```

```python
// Check bucket distribution for stored logs
// Verify data is routing to expected buckets
fetch logs, from: now() - 24h
| summarize {record_count = count()}, by: {dt.system.bucket}
| sort record_count desc
```

```python
// Analyze pipeline processing by source and pipeline
// Shows the relationship between sources and pipelines
fetch logs, from: now() - 24h
| filter isNotNull(dt.openpipeline.pipelines)
| summarize {record_count = count()}, by: {dt.openpipeline.source,
dt.openpipeline.pipelines}
| sort record_count desc
| limit 25
```

```python
// Check for span processing through OpenPipeline
// Spans also flow through OpenPipeline
````
````

```
fetch spans, from: now() - 24h
| summarize {span_count = count()}, by: {dt.openpipeline.source}
| sort span_count desc
```

```python
// View pipeline processing over time
// Helps identify volume patterns and processing trends
fetch logs, from: now() - 24h
| filter isNotNull(dt.openpipeline.pipelines)
| makeTimeseries {record_count = count()}, by: {dt.openpipeline.pipelines},
interval: 1h
```

```python
// Identify logs going to default pipeline (may need custom routing)
// High volume in default may indicate missing routing rules
fetch logs, from: now() - 24h
| filter isNull(dt.openpipeline.pipelines) OR dt.openpipeline.pipelines == "
[]"
| summarize {unrouted_count = count()}, by: {log.source}
| sort unrouted_count desc
| limit 20
```

Understanding Processing Order

The order of processing within a pipeline is critical:

WNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojN2
MzYWVkO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJ
HcmFkaWVudCBpZD0iZXh0R3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+
CiAgICAgIDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMxMGI5ODE7c3RvcC1vc
GFjaXR5OjEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6Iz
A1OTY2OTtzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWF
yR3JhZGllbnQgaWQ9InN0b0dyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUi
PgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojNjM2NmYxO3N0b3Atb
3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOi
M0ZjQ2ZTU7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGZpbHR
lciBpZD0icG9TaGFkb3ciPgogICAgICA8ZmVEcm9wU2hhZG93IGR4PSIxIiBkeT0iMSIgc3RkRGV2
aWF0aW9uPSIyIiBmbG9vZC1vcGFjaXR5PSIwLjE1Ii8+CiAgICA8L2ZpbHRlcj4KICAgIDxtYXJrZ
XIgaWQ9InBvQXJyb3ciIG1hcmtlcldpZHRoPSIxMCIgbWFya2VySGVpZ2h0PSI3IiByZWZYPSI5Ii
ByZWZZPSIzLjUiIG9yaWVudD0iYXV0byI+CiAgICAgIDxwb2x5Z29uIHBvaW50cz0iMCAwLCAxMCA
zLjUsIDAgNyIgZmlsbD0iIzY0NzQ4YiIvPgogICAgPC9tYXJrZXI+CiAgPC9kZWZzPgoKICA8IS0t
IEJhY2tncm91bmQgLS0+CiAgPHJlY3Qgd2lkdGg9IjgwMCIgaGVpZ2h0PSIzNDAiIGZpbGw9IiNmO
GY5ZmEiIHJ4PSIxMCIvPgoKICA8IS0tIFRpdGxlIC0tPgogIDx0ZXh0IHg9IjQwMCIgeT0iMjgiIG
ZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxOCIgZm9udC13ZWlnaHQ
9ImJvbGQiIGZpbGw9IiMzMzMiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlByb2Nlc3NpbmcgU3RhZ2Ug
T3JkZXI8L3RleHQ+CiAgPHRleHQgeD0iNDAwIiB5PSI0OCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzY
W5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmaWxsPSIjNjY2IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj
5Qcm9jZXNzb3JzIGV4ZWN1dGUgaW4gdGhpcyBzcGVjaWZpYyBvcmRlciB3aXRoaW4gdGhlIHBpcGV
saW5lPC90ZXh0PgoKICA8IS0tIFN0YWdlIDE6IE1hc2tpbmcgLS0+CiAgPHJlY3QgeD0iMzAiIHk9
IjcwIiB3aWR0aD0iMTE1IiBoZWlnaHQ9IjkwIiByeD0iOCIgZmlsbD0idXJsKCNtYXNrR3JhZCkiI
GZpbHRlcj0idXJsKCNwb1NoYWRvdykiLz4KICA8dGV4dCB4PSI4NyIgeT0iOTUiIGZvbnQtZmFtaW
x5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZ
pbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4xLiBNQVNLSU5HPC90ZXh0PgogIDx0ZXh0
IHg9Ijg3IiB5PSIxMTUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplP
SIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5TZW
N1cml0eSBGaXJzdCE8L3RleHQ+CiAgPHRleHQgeD0iODciIHk9IjEzMiIgZm9udC1mYW1pbHk9IkF
yaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAu
OSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk1hc2sgcHJvY2Vzc29yIDE8L3RleHQ+CiAgPHRleHQge
D0iODciIHk9IjE0NiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9Ij
EwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk1hc2s
gcHJvY2Vzc29yIDI8L3RleHQ+CgogIDxwYXRoIGQ9Ik0xNDUsMTE1IEwxNjUsMTE1IiBzdHJva2U9
IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCNwb
0Fycm93KSIvPgoKICA8IS0tIFN0YWdlIDI6IEZpbHRlcmluZyAtLT4KICA8cmVjdCB4PSIxNzUiIH
k9IjcwIiB3aWR0aD0iMTE1IiBoZWlnaHQ9IjkwIiByeD0iOCIgZmlsbD0idXJsKCNmaWx0ZXJHcmF
kKSIgZmlsdGVyPSJ1cmwoI3BvU2hhZG93KSIvPgogIDx0ZXh0IHg9IjIzMiIgeT0iOTUiIGZvbnQt
ZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvb
GQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4yLiBGSUxURVJJTkc8L3RleHQ+Ci
AgPHRleHQgeD0iMjMyIiB5PSIxMTUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9
udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlk
ZGxlIj5SZWR1Y2UgVm9sdW1lPC90ZXh0PgogIDx0ZXh0IHg9IjIzMiIgeT0iMTMyIiBmb250LWZhb
WlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NS
wyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RHJvcCBwcm9jZXNzb3IgMTwvdGV4dD4KICA
8dGV4dCB4PSIyMzIiIHk9IjE0NiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250
LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkb

GUiPkRyb3AgcHJvY2Vzc29yIDI8L3RleHQ+CgogIDxwYXRoIGQ9Ik0yOTAsMTE1IEwzMTAsMTE1Ii
BzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0
idXJsKCNwb0Fycm93KSIvPgoKICA8IS0tIFN0YWdlIDM6IFByb2Nlc3NpbmcgLS0+CiAgPHJlY3Qg
eD0iMzIwIiB5PSI3MCIgd2lkdGg9IjExNSIgaGVpZ2h0PSI5MCIgcng9IjgiIGZpbGw9InVybCgjc
HJvY0dyYWQpIiBmaWx0ZXI9InVybCgjcG9TaGFkb3cpIi8+CiAgPHRleHQgeD0iMzc3IiB5PSI5NS
IgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWd
odD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjMuIFBST0NFU1NJTkc8
L3RleHQ+CiAgPHRleHQgeD0iMzc3IiB5PSIxMTUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZ
XJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2
hvcj0ibWlkZGxlIj5UcmFuc2Zvcm08L3RleHQ+CiAgPHRleHQgeD0iMzc3IiB5PSIxMzIiIGZvbnQ
tZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUs
MjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5EUUwgcHJvY2Vzc29yPC90ZXh0PgogI
Dx0ZXh0IHg9IjM3NyIgeT0iMTQ2IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbn
Qtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGR
sZSI+VGVjaCBwYXJzZXI8L3RleHQ+CgogIDxwYXRoIGQ9Ik00MzUsMTE1IEw0NTUsMTE1IiBzdHJv
a2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsK
CNwb0Fycm93KSIvPgoKICA8IS0tIFN0YWdlIDQ6IEVudGl0eSBEZXRlY3Rpb24gLS0+CiAgPHJlY3
QgeD0iNDY1IiB5PSI3MCIgd2lkdGg9IjExNSIgaGVpZ2h0PSI5MCIgcng9IjgiIGZpbGw9InVybCg
jZW50aXR5R3JhZCkiIGZpbHRlcj0idXJsKCNwb1NoYWRvdykiLz4KICA8dGV4dCB4PSI1MjIiIHk9
Ijk1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd
2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+NC4gRU5USVRZIC
90ZXh0PgogIDx0ZXh0IHg9IjUyMiIgeT0iMTEwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2V
yaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNo
b3I9Im1pZGRsZSI+QXV0byBieSBEeW5hdHJhY2U8L3RleHQ+CiAgPHRleHQgeD0iNTIyIiB5PSIxM
zAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icm
diYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5kdC5lbnRpdHkuKjwvdGV
4dD4KICA8dGV4dCB4PSI1MjIiIHk9IjE0NCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlm
IiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yP
SJtaWRkbGUiPmFkZGVkIGhlcmU8L3RleHQ+CgogIDxwYXRoIGQ9Ik01ODAsMTE1IEw2MDAsMTE1Ii
BzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0
idXJsKCNwb0Fycm93KSIvPgoKICA8IS0tIFN0YWdlIDU6IEV4dHJhY3Rpb24gLS0+CiAgPHJlY3Qg
eD0iNjEwIiB5PSI3MCIgd2lkdGg9IjgwIiBoZWlnaHQ9IjkwIiByeD0iOCIgZmlsbD0idXJsKCNle
HRHcmFkKSIgZmlsdGVyPSJ1cmwoI3BvU2hhZG93KSIvPgogIDx0ZXh0IHg9IjY1MCIgeT0iOTUiIG
ZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ
9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj41LiBFWFRSQUNUPC90ZXh0
PgogIDx0ZXh0IHg9IjY1MCIgeT0iMTE1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiI
GZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im
1pZGRsZSI+TWV0cmljczwvdGV4dD4KICA8dGV4dCB4PSI2NTAiIHk9IjEzMCIgZm9udC1mYW1pbHk
9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1
LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkV2ZW50czwvdGV4dD4KICA8dGV4dCB4PSI2NTAiI
Hk9IjE0NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaW
xsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkJpemV2ZW50czw
vdGV4dD4KCiAgPHBhdGggZD0iTTY5MCwxMTUgTDcxMCwxMTUiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ry
b2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI3BvQXJyb3cpIi8+CgogI
DwhLS0gU3RhZ2UgNjogU3RvcmFnZSAtLT4KICA8cmVjdCB4PSI3MjAiIHk9IjcwIiB3aWR0aD0iNj
AiIGhlaWdodD0iOTAiIHJ4PSI4IiBmaWxsPSJ1cmwoI3N0b0dyYWQpIiBmaWx0ZXI9InVybCgjcG9
TaGFkb3cpIi8+CiAgPHRleHQgeD0iNzUwIiB5PSI5NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5z
LXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRle

```
HQtYW5jaG9yPSJtaWRkbGUiPjYuIFNUT1JFPC90ZXh0PgogIDx0ZXh0IHg9Ijc1MCIgeT0iMTIwIi
Bmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmE
oMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+QnVja2V0PC90ZXh0PgogIDx0
ZXh0IHg9Ijc1MCIgeT0iMTM1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc
2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZS
I+Um91dGluZzwvdGV4dD4KCiAgPCEtLSBLZXkgUG9pbnRzIFNlY3Rpb24gLS0+CiAgPHJlY3QgeD0
iMzAiIHk9IjE4MCIgd2lkdGg9Ijc0MCIgaGVpZ2h0PSIxNDUiIHJ4PSI4IiBmaWxsPSIjZmZmIiBz
dHJva2U9IiNlMmU4ZjAiIHN0cm9rZS13aWR0aD0iMiIvPgogIDx0ZXh0IHg9IjQwMCIgeT0iMjA1I
iBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2
h0PSJib2xkIiBmaWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5XaHkgT3JkZXIgTWF0dGV
yczwvdGV4dD4KCiAgPCEtLSBQb2ludCAxIC0tPgogIDxyZWN0IHg9IjUwIiB5PSIyMjAiIHdpZHRo
PSIyMjAiIGhlaWdodD0iOTAiIHJ4PSI2IiBmaWxsPSIjZmVjYWNhIi8+CiAgPHRleHQgeD0iMTYwI
iB5PSIyNDUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm
9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiNkYzI2MjYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk1hc2t
pbmcgRmlyc3Q8L3RleHQ+CiAgPHRleHQgeD0iMTYwIiB5PSIyNjUiIGZvbnQtZmFtaWx5PSJBcmlh
bCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzdmMWQxZCIgdGV4dC1hbmNob3I9I
m1pZGRsZSI+U2Vuc2l0aXZlIGRhdGEgcHJvdGVjdGVkIGV2ZW48L3RleHQ+CiAgPHRleHQgeD0iMT
YwIiB5PSIyODAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCI
gZmlsbD0iIzdmMWQxZCIgdGV4dC1hbmNob3I9Im1pZGRsZSI+aWYgc3Vic2VxdWVudCBwcm9jZXNz
b3JzIGZhaWw8L3RleHQ+CiAgPHRleHQgeD0iMTYwIiB5PSIyOTUiIGZvbnQtZmFtaWx5PSJBcmlhb
Cwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzdmMWQxZCIgdGV4dC1hbmNob3I9Im
1pZGRsZSI+TmV2ZXIgc3RvcmVkIHVubWFza2VkPC90ZXh0PgoKICA8IS0tIFBvaW50IDIgLS0+CiA
gPHJlY3QgeD0iMjkwIiB5PSIyMjAiIHdpZHRoPSIyMjAiIGhlaWdodD0iOTAiIHJ4PSI2IiBmaWxs
PSIjZmVmM2M3Ii8+CiAgPHRleHQgeD0iNDAwIiB5PSIyNDUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc
2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiM5MjQwMG
UiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkRyb3AgRWFybHk8L3RleHQ+CiAgPHRleHQgeD0iNDAwIiB
5PSIyNjUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmls
bD0iIzc4MzUwZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+UmVkdWNlIHZvbHVtZSBiZWZvcmU8L3Rle
HQ+CiAgPHRleHQgeD0iNDAwIiB5PSIyODAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZi
IgZm9udC1zaXplPSIxMCIgZmlsbD0iIzc4MzUwZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+ZXhwZW5
zaXZlIHByb2Nlc3Npbmc8L3RleHQ+CiAgPHRleHQgeD0iNDAwIiB5PSIyOTUiIGZvbnQtZmFtaWx5
PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzc4MzUwZiIgdGV4dC1hb
mNob3I9Im1pZGRsZSI+TmV2ZXIgcGF5IGZvciBkcm9wcGVkIGRhdGE8L3RleHQ+CgogIDwhLS0gUG
9pbnQgMyAtLT4KICA8cmVjdCB4PSI1MzAiIHk9IjIyMCIgd2lkdGg9IjIyMCIgaGVpZ2h0PSI5MCI
gcng9IjYiIGZpbGw9IiNlZGU5ZmUiLz4KICA8dGV4dCB4PSI2NDAiIHk9IjI0NSIgZm9udC1mYW1p
bHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZ
mlsbD0iIzZkMjhkOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RW50aXR5IEFmdGVyIFByb2Nlc3Npbm
c8L3RleHQ+CiAgPHRleHQgeD0iNjQwIiB5PSIyNjUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1
zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzRjMWQ5NSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+
ZHQuZW50aXR5LiogZmllbGRzIHVuYXZhaWxhYmxlPC90ZXh0PgogIDx0ZXh0IHg9IjY0MCIgeT0iM
jgwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9Ii
M0YzFkOTUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPmluIHJvdXRpbmcgb3IgcHJvY2Vzc2luZzwvdGV
4dD4KICA8dGV4dCB4PSI2NDAiIHk9IjI5NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlm
IiBmb250LXNpemU9IjEwIiBmaWxsPSIjNGMxZDk1IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5BdmFpb
GFibGUgaW4gZXh0cmFjdGlvbjwvdGV4dD4KPC9zdmc+Cg==)
```

```
> 💡 **Tip:** Masking is applied FIRST so sensitive data is protected even if
subsequent processors fail.

---

## Summary: Key Architecture Concepts

| Concept | Key Points |
|---------|------------|
| **Data Flow** | Ingest → Route → Process → Extract → Store |
| **Pre-storage** | All processing happens before data is persisted |
| **Routing** | Dynamic routes match data to pipelines |
| **Processing Order** | Masking → Filtering → Processing |
| **Entity Detection** | Happens AFTER processing, BEFORE extraction |
| **Multi-pipeline** | One record can be processed by up to 5 pipelines |
| **DPL** | Powerful pattern language for parsing |
| **Buckets** | Control retention and cost at storage stage |

---

## Next Steps

Now that you understand OpenPipeline architecture, continue with:

| Notebook | Focus Area |
|----------|------------|
| **OPMIG-03** | Migration Assessment & Planning |
| **OPMIG-04** | Pipeline Configuration Fundamentals |
| **OPMIG-05** | Routing & Bucket Management |
| **OPMIG-06** | Processing, Parsing & Transformation |

---

## References

- [OpenPipeline Data Flow](https://docs.dynatrace.com/docs/discover-
dynatrace/platform/openpipeline/concepts/data-flow)
- [OpenPipeline Processing](https://docs.dynatrace.com/docs/discover-
dynatrace/platform/openpipeline/concepts/processing)
- [Dynatrace Pattern Language](https://docs.dynatrace.com/docs/discover-
dynatrace/platform/grail/dynatrace-pattern-language)
- [DPL Architect Tool](https://docs.dynatrace.com/docs/discover-
dynatrace/platform/grail/dynatrace-pattern-language/dpl-architect)

---

*Last Updated: December 12, 2025*
```