🗺️ Service Dependencies & Flow Analysis

> **Series:** SPANS | **Notebook:** 4 of 8 | **Created:** December 2025

Mapping Your Distributed System

This notebook teaches you how to use span data to understand service relationships, analyze request flows, and identify critical dependencies in your system.

Table of Contents

Prerequisites

Before starting this notebook, ensure you have:

- ✅ Completed **SPANS-01** through **SPANS-03**
- ✅ Access to a Dynatrace environment with distributed trace data
- ✅ Understanding of span kinds (server, client, producer, consumer)

1. Understanding Service Topology

Service topology shows how your services connect and communicate:

![Service Topology]

(data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdm
ciIHZpZXdCb3g9IjAgMCA4MDAgMzgwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
iZnJvbnRlbmRHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAg
PHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzYzNjZmMTtzdG9wLW9wYWNpdHk6M
SIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojNGY0NmU1O3
N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWV
udCBpZD0iYmFja2VuZEdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUiPgog
ICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMTQ5NmZmO3N0b3Atb3BhY
2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMwYT
Y0YmM7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckd

```
yYWRpZW50IGlkPSJkYXRhR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+
CiAgICAgIDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMyMmM1NWU7c3RvcC1vc
GFjaXR5OjEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6Iz
E2YTM0YTtzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWF
yR3JhZGllbnQgaWQ9InF1ZXVlR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAw
JSI+CiAgICAgIDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNmOTczMTY7c3Rvc
C1vcGFjaXR5OjEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3
I6I2VhNTgwYztzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8Zml
sdGVyIGlkPSJ0b3BvU2hhZG93Ij4KICAgICAgPGZlRHJvcFNoYWRvdyBkeD0iMiIgZHk9IjIiIHN0
ZERldmlhdGlvbj0iMyIgZmxvb2Qtb3BhY2l0eT0iMC4xNSIvPgogICAgPC9maWx0ZXI+CiAgICA8b
WFya2VyIGlkPSJ0b3BvQXJyb3ciIG1hcmtlcldpZHRoPSI4IiBtYXJrZXJIZWlnaHQ9IjYiIHJlZl
g9IjciIHJlZlk9IjMiIG9yaWVudD0iYXV0byI+CiAgICAgIDxwb2x5Z29uIHBvaW50cz0iMCAwLCA
4IDMsIDAgNiIgZmlsbD0iIzY0NzQ4YiIvPgogICAgPC9tYXJrZXI+CiAgPC9kZWZzPgoKICA8IS0t
IEJhY2tncm91bmQgLS0+CiAgPHJlY3Qgd2lkdGg9IjgwMCIgaGVpZ2h0PSIzODAiIGZpbGw9IiNmO
GY5ZmEiIHJ4PSIxMCIvPgoKICA8IS0tIFRpdGxlIC0tPgogIDx0ZXh0IHg9IjQwMCIgeT0iMjgiIG
ZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxOCIgZm9udC13ZWlnaHQ
9ImJvbGQiIGZpbGw9IiMzMzMiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlNlcnZpY2UgVG9wb2xvZ3kg
ZnJvbSBTcGFuIERhdGE8L3RleHQ+CgogIDwhLS0gTGF5ZXIgTGFiZWxzIChsZWZ0IHNpZGUpIC0tP
gogIDxyZWN0IHg9IjIwIiB5PSI2NSIgd2lkdGg9IjkwIiBoZWlnaHQ9IjI0IiByeD0iNCIgZmlsbD
0iIzYzNjZmMSIvPgogIDx0ZXh0IHg9IjY1IiB5PSI4MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5
zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRl
eHQtYW5jaG9yPSJtaWRkbGUiPkZST05URU5EPC90ZXh0PgoKICA8cmVjdCB4PSIyMCIgeT0iMTU1I
iB3aWR0aD0iOTAiIGhlaWdodD0iMjQiIHJ4PSI0IiBmaWxsPSIjMTQ5NmZmIi8+CiAgPHRleHQgeD
0iNjUiIHk9IjE3MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE
xIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkJB
Q0tFTkQ8L3RleHQ+CgogIDxyZWN0IHg9IjIwIiB5PSIyNzUiIHdpZHRoPSI5MCIgaGVpZ2h0PSIyN
CIgcng9IjQiIGZpbGw9IiMyMmM1NWUiLz4KICA8dGV4dCB4PSI2NSIgeT0iMjkyIiBmb250LWZhbW
lseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiB
maWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+REFUQTwvdGV4dD4KCiAgPCEtLSBGcm9u
dGVuZCBMYXllciAtLT4KICA8cmVjdCB4PSIxMzAiIHk9IjU1IiB3aWR0aD0iMTIwIiBoZWlnaHQ9I
jUwIiByeD0iOCIgZmlsbD0idXJsKCNmcm9udGVuZEdyYWQpIiBmaWx0ZXI9InVybCgjdG9wb1NoYW
RvdykiLz4KICA8dGV4dCB4PSIxOTAiIHk9Ijg1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2V
yaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1h
bmNob3I9Im1pZGRsZSI+V2ViIEFwcDwvdGV4dD4KCiAgPHJlY3QgeD0iMjcwIiB5PSI1NSIgd2lkd
Gg9IjEyMCIgaGVpZ2h0PSI1MCIgcng9IjgiIGZpbGw9InVybCgjZnJvbnRlbmRHcmFkKSIgZmlsdG
VyPSJ1cmwoI3RvcG9TaGFkb3cpIi8+CiAgPHRleHQgeD0iMzMwIiB5PSI4NSIgZm9udC1mYW1pbHk
9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmls
bD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk1vYmlsZSBBUEk8L3RleHQ+CgogIDwhLS0gQ
mFja2VuZCBMYXllciAtLT4KICA8cmVjdCB4PSIxMzAiIHk9IjE0NSIgd2lkdGg9IjExMCIgaGVpZ2
h0PSI1MCIgcng9IjgiIGZpbGw9InVybCgjYmFja2VuZEdyYWQpIiBmaWx0ZXI9InVybCgjdG9wb1N
oYWRvdykiLz4KICA8dGV4dCB4PSIxODUiIHk9IjE3NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5z
LXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRle
HQtYW5jaG9yPSJtaWRkbGUiPkFQSSBHYXRld2F5PC90ZXh0PgoKICA8cmVjdCB4PSIyNjAiIHk9Ij
E0NSIgd2lkdGg9IjExMCIgaGVpZ2h0PSI1MCIgcng9IjgiIGZpbGw9InVybCgjYmFja2VuZEdyYWQ
pIiBmaWx0ZXI9InVybCgjdG9wb1NoYWRvdykiLz4KICA8dGV4dCB4PSIzMTUiIHk9IjE3NSIgZm9u
dC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iY
m9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlVzZXIgU2VydmljZTwvdGV4dD
4KCiAgPHJlY3QgeD0iMzkwIiB5PSIxNDUiIHdpZHRoPSIxMTAiIGhlaWdodD0iNTAiIHJ4PSI4IiB
```

```
maWxsPSJ1cmwoI2JhY2tlbmRHcmFkKSIgZmlsdGVyPSJ1cmwoI3RvcG9TaGFkb3cpIi8+CiAgPHRl
eHQgeD0iNDQ1IiB5PSIxNzUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1za
XplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZG
xlIj5PcmRlciBTZXJ2aWNlPC90ZXh0PgoKICA8cmVjdCB4PSI1MjAiIHk9IjE0NSIgd2lkdGg9IjE
xMCIgaGVpZ2h0PSI1MCIgcng9IjgiIGZpbGw9InVybCgjYmFja2VuZEdyYWQpIiBmaWx0ZXI9InVy
bCgjdG9wb1NoYWRvdykiLz4KICA8dGV4dCB4PSI1NzUiIHk9IjE3NSIgZm9udC1mYW1pbHk9IkFya
WFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2
hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlBheW1lbnQgU3ZjPC90ZXh0PgoKICA8IS0tIERhdGE
gTGF5ZXIgLS0+CiAgPHJlY3QgeD0iMjYwIiB5PSIyNjUiIHdpZHRoPSIxMDAiIGhlaWdodD0iNDUi
IHJ4PSI4IiBmaWxsPSJ1cmwoI2RhdGFHcmFkKSIgZmlsdGVyPSJ1cmwoI3RvcG9TaGFkb3cpIi8+C
iAgPHRleHQgeD0iMzEwIiB5PSIyOTMiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm
9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0
ibWlkZGxlIj5Qb3N0Z3JlU1FMPC90ZXh0PgoKICA8cmVjdCB4PSIzODAiIHk9IjI2NSIgd2lkdGg9
IjEwMCIgaGVpZ2h0PSI0NSIgcng9IjgiIGZpbGw9InVybCgjZGF0YUdyYWQpIiBmaWx0ZXI9InVyb
CgjdG9wb1NoYWRvdykiLz4KICA8dGV4dCB4PSI0MzAiIHk9IjI5MyIgZm9udC1mYW1pbHk9IkFyaW
FsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2h
pdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk1vbmdvREI8L3RleHQ+CgogIDxyZWN0IHg9IjUwMCIg
eT0iMjY1IiB3aWR0aD0iMTAwIiBoZWlnaHQ9IjQ1IiByeD0iOCIgZmlsbD0idXJsKCNkYXRhR3JhZ
CkiIGZpbHRlcj0idXJsKCN0b3BvU2hhZG93KSIvPgogIDx0ZXh0IHg9IjU1MCIgeT0iMjkzIiBmb2
50LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJ
ib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+UmVkaXM8L3RleHQ+CgogIDwh
LS0gUXVldWUgLS0+CiAgPHJlY3QgeD0iNjUwIiB5PSIxNDUiIHdpZHRoPSIxMTAiIGhlaWdodD0iN
TAiIHJ4PSI4IiBmaWxsPSJ1cmwoI3F1ZXVlR3JhZCkiIGZpbHRlcj0idXJsKCN0b3BvU2hhZG93KS
IvPgogIDx0ZXh0IHg9IjcwNSIgeT0iMTc1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWY
iIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNo
b3I9Im1pZGRsZSI+S2Fma2E8L3RleHQ+CgogIDwhLS0gQXJyb3dzIC0gRnJvbnRlbmQgdG8gQVBJI
EdhdGV3YXkgLS0+CiAgPHBhdGggZD0iTTE5MCwxMDUgTDE5MCwxNDAiIHN0cm9rZT0iIzY0NzQ4Yi
Igc3Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI3RvcG9BcnJvdyk
iLz4KICA8cGF0aCBkPSJNMzMwLDEwNSBMMjMwLDE0MCIgc3Ryb2tlPSIjNjQ3NDhiIiBzdHJva2Ut
d2lkdGg9IjIiIGZpbGw9Im5vbmUiIG1hcmtlci1lbmQ9InVybCgjdG9wb0Fycm93KSIvPgoKICA8I
S0tIEFycm93cyAtIEFQSSBHYXRld2F5IHRvIFNlcnZpY2VzIChjdXJ2ZWQgdG8gYXZvaWQgb3Zlcm
xhcCkgLS0+CiAgPHBhdGggZD0iTTI0MCwxNzAgUTI1MCwxNzAgMjU1LDE3MCIgc3Ryb2tlPSIjNjQ
3NDhiIiBzdHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmUiIG1hcmtlci1lbmQ9InVybCgjdG9wb0Fy
cm93KSIvPgogIDxwYXRoIGQ9Ik0yNDAsMTc1IFEzMjAsMjEwIDM4NSwxNzUiIHN0cm9rZT0iIzY0N
zQ4YiIgc3Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI3RvcG9Bcn
JvdykiLz4KCiAgPCEtLSBBcnJvd3MgLSBTZXJ2aWNlcyB0byBTZXJ2aWNlUgLS0+CiAgPHBhdGggZD0
iTTM3MCwxNzAgTDM4NSwxNzAiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ryb2tlLXdpZHRoPSIyIiBmaWxs
PSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI3RvcG9BcnJvdykiLz4KICA8cGF0aCBkPSJNNTAwLDE3M
CBMNTE1LDE3MCIgc3Ryb2tlPSIjNjQ3NDhiIiBzdHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmUiIG
1hcmtlci1lbmQ9InVybCgjdG9wb0Fycm93KSIvPgoKICA8IS0tIEFycm93cyAtIFNlcnZpY2VzIHR
vIERhdGEgLS0+CiAgPHBhdGggZD0iTTMxNSwxOTUgTDMxMiwyNjAiIHN0cm9rZT0iIzY0NzQ4YiIg
c3Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI3RvcG9BcnJvdykiL
z4KICA8cGF0aCBkPSJNNDQ1LDE5NSBMNDMyLDI2MCIgc3Ryb2tlPSIjNjQ3NDhiIiBzdHJva2Utd2
lkdGg9IjIiIGZpbGw9Im5vbmUiIG1hcmtlci1lbmQ9InVybCgjdG9wb0Fycm93KSIvPgogIDxwYXR
oIGQ9Ik01NzUsMTk1IEw1NTIsMjYwIiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIg
ZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCN0b3BvQXJyb3cpIi8+CgogIDwhLS0gQXJyb3cgd
G8gS2Fma2EgLS0+CiAgPHBhdGggZD0iTTYzMCwxNzAgTDY0NSwxNzAiIHN0cm9rZT0iIzY0NzQ4Yi
Igc3Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI3RvcG9BcnJvdyk
```

```
iLz4KCiAgPCEtLSBMZWdlbmQgYm94IC0tPgogIDxyZWN0IHg9IjYyMCIgeT0iMjY1IiB3aWR0aD0i
MTYwIiBoZWlnaHQ9Ijk1IiByeD0iNiIgZmlsbD0iI2ZmZiIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva
2Utd2lkdGg9IjIiLz4KICA8dGV4dCB4PSI3MDAiIHk9IjI4NyIgZm9udC1mYW1pbHk9IkFyaWFsLC
BzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzMzMyI
gdGV4dC1hbmNob3I9Im1pZGRsZSI+RGlzY292ZXJlZCB2aWEgRFFMPC90ZXh0PgogIDx0ZXh0IHg9
IjYzNSIgeT0iMzA3IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsP
SIjNjY2Ij5mZXRjaCBzcGFuczwvdGV4dD4KICA8dGV4dCB4PSI2MzUiIHk9IjMyMSIgZm9udC1mYW
1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY2NiI+fCBmaWx0ZXIgc3Bhbi5
raW5kPC90ZXh0PgogIDx0ZXh0IHg9IjY0NSIgeT0iMzM1IiBmb250LWZhbWlseT0ibW9ub3NwYWNl
IiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjY2Ij4gID09ICJjbGllbnQiPC90ZXh0PgogIDx0ZXh0I
Hg9IjYzNSIgeT0iMzQ5IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaW
xsPSIjNjY2Ij58IHN1bW1hcml6ZSBieTp7Li4ufTwvdGV4dD4KPC9zdmc+Cg==)
```

~~~
### Key Span Types for Topology

| Span Kind | Role | What It Tells You |
|-----------|------|-------------------|
| `server` | Receives requests | Entry points, inbound traffic |
| `client` | Makes requests | Outbound calls, dependencies |
| `producer` | Sends messages | Async event sources |
| `consumer` | Receives messages | Async event handlers |

---

## 2. Discovering Services

First, discover all services generating spans in your environment:

```dql
// List all services with span counts
fetch spans
| summarize {
    span_count = count(),
    operations = countDistinct(span.name)
  }, by: {service.name}
| sort span_count desc
| limit 30
```

```dql
// Service role analysis - understand each service's function
fetch spans
| summarize {
    server_spans = countIf(span.kind == "server"),
    client_spans = countIf(span.kind == "client"),
    internal_spans = countIf(span.kind == "internal"),
~~~

~~~
    producer_spans = countIf(span.kind == "producer"),
    consumer_spans = countIf(span.kind == "consumer")
  }, by: {service.name}
| sort server_spans desc
| limit 30
```

```dql
// Discover all operations/endpoints per service
fetch spans
| filter span.kind == "server"
| summarize {
    request_count = count(),
    avg_duration_ms = avg(duration) / 1000000
  }, by: {service.name, span.name}
| sort service.name asc, request_count desc
| limit 50
```

---

## 3. Mapping Service Dependencies

Use CLIENT spans to understand which services call which other services:

### Key Attributes for Dependencies

| Attribute | Description |
|-----------|-------------|
| `peer.service` | Target service name (if instrumented) |
| `server.address` | Target host/address |
| `server.port` | Target port |
| `http.host` | HTTP host header |

```dql
// Map service-to-service calls using CLIENT spans
fetch spans
| filter span.kind == "client"
| summarize {
    call_count = count(),
    avg_latency_ms = avg(duration) / 1000000,
    error_count = countIf(span.status_code == "error")
  }, by: {service.name, span.name}
| fieldsAdd error_rate_pct = (error_count * 100.0) / call_count
| sort call_count desc
| limit 50
```
~~~

```dql
// Find services called by other services using peer.service attribute
// peer.service shows the target service name when available
fetch spans
| filter span.kind == "client" and isNotNull(peer.service)
| summarize {
    call_count = count(),
    avg_latency_ms = avg(duration) / 1000000
  }, by: {service.name, peer.service}
| sort call_count desc
| limit 30
```

```dql
// Map dependencies using HTTP host/URL information
fetch spans
| filter span.kind == "client" and isNotNull(server.address)
| summarize {
    call_count = count(),
    avg_latency_ms = avg(duration) / 1000000,
    error_count = countIf(span.status_code == "error")
  }, by: {service.name, server.address}
| fieldsAdd error_rate_pct = (error_count * 100.0) / call_count
| sort call_count desc
| limit 30
```

---

4. Client-Server Call Patterns

Analyze the matching CLIENT and SERVER span pairs to understand inter-service communication:

```dql
// Analyze outbound calls from each service
fetch spans
| filter span.kind == "client"
| summarize {
    outbound_calls = count(),
    avg_latency_ms = avg(duration) / 1000000,
    p99_latency_ms = percentile(duration, 99) / 1000000,
    error_count = countIf(span.status_code == "error")
  }, by: {service.name}
| fieldsAdd error_rate_pct = (error_count * 100.0) / outbound_calls
| sort outbound_calls desc
| limit 20
```

~~~
```dql
// Analyze inbound requests to each service
fetch spans
| filter span.kind == "server"
| summarize {
    inbound_requests = count(),
    avg_latency_ms = avg(duration) / 1000000,
    p99_latency_ms = percentile(duration, 99) / 1000000,
    error_count = countIf(span.status_code == "error")
  }, by: {service.name}
| fieldsAdd error_rate_pct = (error_count * 100.0) / inbound_requests
| sort inbound_requests desc
| limit 20
```

```dql
// Find services with high outbound/inbound ratio (integration heavy)
fetch spans
| summarize {
    inbound = countIf(span.kind == "server"),
    outbound = countIf(span.kind == "client")
  }, by: {service.name}
| filter inbound > 0
| fieldsAdd outbound_ratio = outbound / inbound
| sort outbound_ratio desc
| limit 20
```

---

## 5. Async Messaging Flows

Analyze asynchronous messaging patterns using PRODUCER and CONSUMER spans:

![Async Messaging Flow]
(data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdm
ciIHZpZXdCb3g9IjAgMCA4MDAgMzIwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
icHJvZHVjZXJHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAg
PHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzYzNjZmMTtzdG9wLW9wYWNpdHk6M
SIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojNGY0NmU1O3
N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWV
udCBpZD0icXVldWVHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAg
ICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2Y5NzMxNjtzdG9wLW9wYWNpd
Hk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZWE1OD
BjO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmF
kaWVudCBpZD0iY29uc3VtZXJHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAl
Ij4KICAgICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzIyYzU1ZTtzdG9wL
~~~

```
W9wYWNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcj
ojMTZhMzRhO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5
lYXJHcmFkaWVudCBpZD0idHJhY2VHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIw
JSI+CiAgICAgIDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM4YjVjZjY7c3Rvc
C1vcGFjaXR5OjAuMyIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2
xvcjojN2MzYWVkO3N0b3Atb3BhY2l0eTowLjMiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICA
gPGZpbHRlciBpZD0ibXNnU2hhZG93Ij4KICAgICAgPGZlRHJvcFNoYWRvdyBkeD0iMiIgZHk9IjIi
IHN0ZERldmlhdGlvbj0iMyIgZmxvb2Qtb3BhY2l0eT0iMC4xNSIvPgogICAgPC9maWx0ZXI+CiAgI
CA8bWFya2VyIGlkPSJtc2dBcnJvdyIgbWFya2VyV2lkdGg9IjEwIiBtYXJrZXJIZWlnaHQ9IjciIH
JlZlg9IjkiIHJlZlk9IjMuNSIgb3JpZW50PSJhdXRvIj4KICAgICAgPHBvbHlnb24gcG9pbnRzPSI
wIDAsIDEwIDMuNSwgMCA3IiBmaWxsPSIjNjQ3NDhiIi8+CiAgICA8L21hcmtlcj4KICAgIDxtYXJr
ZXIgaWQ9Im1zZ0Fycm93T3JhbmdlIiBtYXJrZXJXaWR0aD0iMTAiIG1hcmtlckhlaWdodD0iNyIgc
mVmWD0iOSIgcmVmWT0iMy41IiBvcmllbnQ9ImF1dG8iPgogICAgICA8cG9seWdvbiBwb2ludHM9Ij
AgMCwgMTAgMy41LCAwIDciIGZpbGw9IiNmOTczMTYiLz4KICAgIDwvbWFya2VyPgogIDwvZGVmcz4
KCiAgPCEtLSBCYWNrZ3JvdW5kIC0tPgogIDxyZWN0IHdpZHRoPSI4MDAiIGhlaWdodD0iMzIwIiBm
aWxsPSIjZjhmOWZhIiByeD0iMTAiLz4KCiAgPCEtLSBUaXRsZSAtLT4KICA8dGV4dCB4PSI0MDAiI
Hk9IjI4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTgiIGZvbn
Qtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Bc3luYyBNZXN
zYWdpbmcgRmxvdyAoUHJvZHVjZXIvQ29uc3VtZXIgU3BhbnMpPC90ZXh0PgoKICA8IS0tIFRyYWNl
IGNvbnRleHQgYmFja2dyb3VuZCAtLT4KICA8cmVjdCB4PSIzMCIgeT0iNjAiIHdpZHRoPSI3NDAiI
GhlaWdodD0iOTAiIHJ4PSI4IiBmaWxsPSJ1cmwoI3RyYWNlR3JhZCkiIHN0cm9rZT0iIzhiNWNmNi
Igc3Ryb2tlLXdpZHRoPSIxIiBzdHJva2UtZGFzaGFycmF5PSI1LDUiLz4KICA8dGV4dCB4PSI0MDA
iIHk9IjgwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZp
bGw9IiM3YzNhZWQiIHRleHQtYW5jaG9yPSJtaWRkbGUiPnRyYWNlLmlkOiBhYmMxMjMgKGNvbnRle
HQgcHJvcGFnYXRlZCB0aHJvdWdoIG1lc3NhZ2UgaGVhZGVycyk8L3RleHQ+CgogIDwhLS0gUHJvZH
VjZXIgU2VydmljZSAtLT4KICA8cmVjdCB4PSI1MCIgeT0iOTUiIHdpZHRoPSIxNTAiIGhlaWdodD0
iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwoI3Byb2R1Y2VyR3JhZCkiIGZpbHRlcj0idXJsKCNtc2dTaGFk
b3cpIi8+CiAgPHRleHQgeD0iMTI1IiB5PSIxMTUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZ
XJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LW
FuY2hvcj0ibWlkZGxlIj5QYXltZW50IFNlcnZpY2U8L3RleHQ+CiAgPHRleHQgeD0iMTI1IiB5PSI
xMzAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0i
cmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5QUk9EVUNFUiBzcGFuP
C90ZXh0PgoKICA8IS0tIEFycm93IHRvIFF1ZXVlIC0tPgogIDxwYXRoIGQ9Ik0yMDAsMTE3IEwyOD
AsMTE3IiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2V
yLWVuZD0idXJsKCNtc2dBcnJvdykiLz4KICA8dGV4dCB4PSIyNDAiIHk9IjEwOCIgZm9udC1mYW1p
bHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIiB0ZXh0L
WFuY2hvcj0ibWlkZGxlIj5wdWJsaXNoPC90ZXh0PgoKICA8IS0tIE1lc3NhZ2UgUXVldWUgLS0+Ci
AgPHJlY3QgeD0iMjg1IiB5PSI4NSIgd2lkdGg9IjE4MCIgaGVpZ2h0PSI2NSIgcng9IjgiIGZpbGw
9InVybCgjcXVldWVHcmFkKSIgZmlsdGVyPSJ1cmwoI21zZ1NoYWRvdykiLz4KICA8dGV4dCB4PSIz
NzUiIHk9IjExMCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyI
iBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkthZm
thIFRvcGljPC90ZXh0PgogIDx0ZXh0IHg9IjM3NSIgeT0iMTI4IiBmb250LWZhbWlseT0iQXJpYWw
sIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIg
dGV4dC1hbmNob3I9Im1pZGRsZSI+b3JkZXItY29tcGxldGVkPC90ZXh0PgogIDx0ZXh0IHg9IjM3N
SIgeT0iMTQzIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2
JhKDI1NSwyNTUsMjU1LDAuOCkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPm1lc3NhZ2luZy5kZXN0aW5
hdGlvbi5uYW1lPC90ZXh0PgoKICA8IS0tIEFycm93IGZyb20gUXVldWUgdG8gQ29uc3VtZXIgLS0+
CiAgPHBhdGggZD0iTTQ2NSwxMTcgTDU0NSwxMTciIHN0cm9rZT0iIzY0NzQ4YiIgc3Ryb2tlLXdpZ
```

```
HRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI21zZ0Fycm93KSIvPgogIDx0ZXh0IH
g9IjUwNSIgeT0iMTA4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0
iMTAiIGZpbGw9IiM2NDc0OGIiIHRleHQtYW5jaG9yPSJtaWRkbGUiPnJlY2VpdmU8L3RleHQ+Cgog
IDwhLS0gQ29uc3VtZXIgU2VydmljZSAtLT4KICA8cmVjdCB4PSI1NTAiIHk9Ijk1IiB3aWR0aD0iM
TUwIiBoZWlnaHQ9IjQ1IiByeD0iOCIgZmlsbD0idXJsKCNjb25zdW1lckdyYWQpIiBmaWx0ZXI9In
VybCgjbXNnU2hhZG93KSIvPgogIDx0ZXh0IHg9IjYyNSIgeT0iMTE1IiBmb250LWZhbWlseT0iQXJ
pYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3
aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RW1haWwgU2VydmljZTwvdGV4dD4KICA8dGV4dCB4P
SI2MjUiIHk9IjEzMCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9Ij
EwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkNPTlN
VTUVSIHNwYW48L3RleHQ+CgogIDwhLS0gU2Vjb25kIENvbnN1bWVyIChmYW4tb3V0IHBhdHRlcm4p
IC0tPgogIDxwYXRoIGQ9Ik00NjUsMTI1IEw1NDUsMTgwIiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZ
S13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCNtc2dBcnJvdykiLz4KICA8cm
VjdCB4PSI1NTAiIHk9IjE2NSIgd2lkdGg9IjE1MCIgaGVpZ2h0PSI0NSIgcng9IjgiIGZpbGw9InV
ybCgjY29uc3VtZXJHcmFkKSIgZmlsdGVyPSJ1cmwoI21zZ1NoYWRvdykiLz4KICA8dGV4dCB4PSI2
MjUiIHk9IjE4NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExI
iBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkFuYW
x5dGljcyBTZXJ2aWNlPC90ZXh0PgogIDx0ZXh0IHg9IjYyNSIgeT0iMjAwIiBmb250LWZhbWlseT0
iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUs
MC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+Q09OU1VNRVIgc3BhbjwvdGV4dD4KCiAgPCEtLSBLZ
XkgRmllbGRzIFNlY3Rpb24gLS0+CiAgPHJlY3QgeD0iNTAiIHk9IjIzMCIgd2lkdGg9IjMzMCIgaG
VpZ2h0PSI3NSIgcng9IjYiIGZpbGw9IiNmZmYiIHN0cm9rZT0iI2UyZThmMCIgc3Ryb2tlLXdpZHR
oPSIyIi8+CiAgPHRleHQgeD0iMjE1IiB5PSIyNTAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1z
ZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiMzMzMiIHRleHQtY
W5jaG9yPSJtaWRkbGUiPktleSBNZXNzYWdpbmcgRmllbGRzPC90ZXh0PgogIDx0ZXh0IHg9IjY1Ii
B5PSIyNzAiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM2NjY
iPnNwYW4ua2luZDogInByb2R1Y2VyIiB8ICJjb25zdW1lciI8L3RleHQ+CiAgPHRleHQgeD0iNjUi
IHk9IjI4NSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY2N
iI+bWVzc2FnaW5nLnN5c3RlbTogImthZmthIjwvdGV4dD4KICA8dGV4dCB4PSI2NSIgeT0iMzAwIi
Bmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjY2Ij5tZXNzYWd
pbmcuZGVzdGluYXRpb24ubmFtZTogIm9yZGVyLWNvbXBsZXRlZCI8L3RleHQ+CgogIDwhLS0gRFFM
IFF1ZXJ5IFNlY3Rpb24gLS0+CiAgPHJlY3QgeD0iNDAwIiB5PSIyMzAiIHdpZHRoPSIzNzAiIGhla
WdodD0iNzUiIHJ4PSI2IiBmaWxsPSIjMWUyOTNiIi8+CiAgPHRleHQgeD0iNTg1IiB5PSIyNTAiIG
ZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ
9ImJvbGQiIGZpbGw9IiM5NGEzYjgiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlF1ZXJ5IEFzeW5jIEZs
b3dzPC90ZXh0PgogIDx0ZXh0IHg9IjQxNSIgeT0iMjcwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlI
iBmb250LXNpemU9IjEwIiBmaWxsPSIjMjJjNTVlIj5mZXRjaDwvdGV4dD4KICA8dGV4dCB4PSI0NT
UiIHk9IjI3MCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2Y
4ZmFmYyI+c3BhbnM8L3RleHQ+CiAgPHRleHQgeD0iNDE1IiB5PSIyODUiIGZvbnQtZmFtaWx5PSJt
b25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiMxNDk2ZmYiPnwgZmlsdGVyPC90ZXh0PgogI
Dx0ZXh0IHg9IjQ3NSIgeT0iMjg1IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9Ij
EwIiBmaWxsPSIjZjhmYWZjIj5pbihzcGFuLmtpbmQsIHsicHJvZHVjZXIiLCAiY29uc3VtZXIifSk
8L3RleHQ+CiAgPHRleHQgeD0iNDE1IiB5PSIzMDAiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZv
bnQtc2l6ZT0iMTAiIGZpbGw9IiMxNDk2ZmYiPnwgc3VtbWFyaXplPC90ZXh0PgogIDx0ZXh0IHg9I
jQ5NSIgeT0iMzAwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPS
IjZjhmYWZjIj5jb3VudCgpLCBieTp7bWVzc2FnaW5nLmRlc3RpbmF0aW9uLm5hbWV9PC90ZXh0Pgo
8L3N2Zz4K)
```

~~~
### Key Messaging Attributes

| Attribute | Description |
|-----------|-------------|
| `messaging.system` | Kafka, RabbitMQ, etc. |
| `messaging.destination.name` | Topic/queue name |
| `messaging.operation` | publish, receive, etc. |

```dql
// Find message producers (services sending async messages)
fetch spans
| filter span.kind == "producer"
| summarize {
    messages_sent = count(),
    avg_duration_ms = avg(duration) / 1000000,
    error_count = countIf(span.status_code == "error")
  }, by: {service.name, span.name}
| sort messages_sent desc
| limit 20
```

```dql
// Find message consumers (services receiving async messages)
fetch spans
| filter span.kind == "consumer"
| summarize {
    messages_received = count(),
    avg_processing_ms = avg(duration) / 1000000,
    error_count = countIf(span.status_code == "error")
  }, by: {service.name, span.name}
| fieldsAdd error_rate_pct = (error_count * 100.0) / messages_received
| sort messages_received desc
| limit 20
```

```dql
// Analyze messaging system usage (Kafka, RabbitMQ, etc.)
fetch spans
| filter span.kind == "producer" or span.kind == "consumer"
| filter isNotNull(messaging.system)
| summarize {
    message_count = count(),
    producers = countIf(span.kind == "producer"),
    consumers = countIf(span.kind == "consumer"),
    error_count = countIf(span.status_code == "error")
  }, by: {messaging.system, messaging.destination.name}
~~~

~~~
| sort message_count desc
| limit 20
```

```dql
// Map producer to consumer relationships
fetch spans
| filter span.kind == "producer" or span.kind == "consumer"
| summarize {
    span_count = count()
  }, by: {service.name, span.kind, messaging.destination.name}
| sort messaging.destination.name, span.kind
| limit 30
```

---

## 6. Trace Hierarchy Analysis

Analyze parent-child relationships within traces to understand call depth:

```dql
// Count spans per trace to understand trace complexity
fetch spans
| summarize {
    span_count = count(),
    services_involved = countDistinct(service.name),
    total_duration_ms = sum(duration) / 1000000
  }, by: {trace.id}
| sort span_count desc
| limit 25
```

```dql
// Examine a complete trace hierarchy
// Replace YOUR_TRACE_ID with an actual trace ID from above
fetch spans
// | filter trace.id == "YOUR_TRACE_ID"
| fieldsAdd duration_ms = duration / 1000000
| fields start_time,
         span.id,
         span.parent_id,
         service.name,
         span.name,
         span.kind,
         duration_ms
| sort start_time asc
| limit 100
~~~

~~~
```

```dql
// Find entry points (root spans) and their downstream services
fetch spans
| filter isNull(span.parent_id)
| summarize {
    entry_count = count(),
    avg_duration_ms = avg(duration) / 1000000
  }, by: {service.name, span.name}
| sort entry_count desc
| limit 20
```

---

## 7. Cross-Service Latency Analysis

Identify latency hot spots between services:

```dql
// Latency by service-to-service call
fetch spans
| filter span.kind == "client" and isNotNull(server.address)
| summarize {
    call_count = count(),
    avg_ms = avg(duration) / 1000000,
    p95_ms = percentile(duration, 95) / 1000000,
    p99_ms = percentile(duration, 99) / 1000000
  }, by: {service.name, server.address}
| sort p95_ms desc
| limit 20
```

```dql
// Time spent per service in traces
fetch spans
| summarize {
    total_time_ms = sum(duration) / 1000000,
    span_count = count(),
    avg_per_span_ms = avg(duration) / 1000000
  }, by: {service.name}
| sort total_time_ms desc
| limit 20
```

```dql
// Find slowest dependencies (CLIENT spans)
~~~

~~~~
fetch spans
| filter span.kind == "client"
| filter duration > 500ms
| summarize {
    slow_call_count = count(),
    avg_duration_ms = avg(duration) / 1000000,
    max_duration_ms = max(duration) / 1000000
  }, by: {service.name, span.name}
| sort avg_duration_ms desc
| limit 20
```

---

## 8. Critical Path Analysis

Identify the services and operations that contribute most to end-to-end
latency:

![Critical Path Analysis]
~~~~

~~~~
(data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdm
ciIHZpZXdCb3g9IjAgMCA4MDAgMzQwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
iZmFzdEdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjAlIj4KICAgICAgPHN0b3Ag
b2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzIyYzU1ZTtzdG9wLW9wYWNpdHk6MSIgLz4KI
CAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMTZhMzRhO3N0b3Atb3
BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
ic2xvd0dyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjAlIj4KICAgICAgPHN0b3Ag
b2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2VmNDQ0NDtzdG9wLW9wYWNpdHk6MSIgLz4KI
CAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZGMyNjI2O3N0b3Atb3
BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
ibWVkR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMCUiPgogICAgICA8c3RvcCBv
ZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZjU5ZTBiO3N0b3Atb3BhY2l0eToxIiAvPgogI
CAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNkOTc3MDY7c3RvcC1vcG
FjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWRpZW50IGlkPSJ
ub3JtYWxHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIwJSI+CiAgICAgIDxzdG9w
IG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMxNDk2ZmY7c3RvcC1vcGFjaXR5OjEiIC8+C
iAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzBhNjRiYztzdG9wLW
9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8ZmlsdGVyIGlkPSJjcFNoYWR
vdyI+CiAgICAgIDxmZURyb3BTaGFkb3cgZHg9IjEiIGR5PSIxIiBzdGREZXZpYXRpb249IjIiIGZs
b29kLW9wYWNpdHk9IjAuMTUiLz4KICAgIDwvZmlsdGVyPgogIDwvZGVmcz4KCiAgPCEtLSBCYWNrZ
3JvdW5kIC0tPgogIDxyZWN0IHdpZHRoPSI4MDAiIGhlaWdodD0iMzQwIiBmaWxsPSIjZjhmOWZhIi
ByeD0iMTAiLz4KCiAgPCEtLSBUaXRsZSAtLT4KICA8dGV4dCB4PSI0MDAiIHk9IjI4IiBmb250LWZ
hbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTgiIGZvbnQtd2VpZ2h0PSJib2xk
IiBmaWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Dcml0aWNhbCBQYXRoIEFuYWx5c2lzP
C90ZXh0PgogIDx0ZXh0IHg9IjQwMCIgeT0iNDgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZX
JpZiIgZm9udC1zaXplPSIxMiIgZmlsbD0iIzY2NiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+VG90YWw
gVHJhY2UgRHVyYXRpb246IDUwMG1zIHwgSWRlbnRpZnkgdGhlIGJvdHRsZW5lY2sgdG8gb3B0aW1p
emU8L3RleHQ+CgogIDwhLS0gVGltZWxpbmUgaGVhZGVyIC0tPgogIDx0ZXh0IHg9IjE2MCIgeT0iN
~~~~

```
zgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsbD0iIz
Y0NzQ4YiI+MG1zPC90ZXh0PgogIDx0ZXh0IHg9IjM2MCIgeT0iNzgiIGZvbnQtZmFtaWx5PSJBcml
hbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsbD0iIzY0NzQ4YiIgdGV4dC1hbmNob3I9
Im1pZGRsZSI+MjUwbXM8L3RleHQ+CiAgPHRleHQgeD0iNTYwIiB5PSI3OCIgZm9udC1mYW1pbHk9I
kFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjNjQ3NDhiIiB0ZXh0LWFuY2
hvcj0iZW5kIj41MDBtczwvdGV4dD4KCiAgPCEtLSBUaW1lbGluZSBiYWNrZ3JvdW5kIC0tPgogIDx
yZWN0IHg9IjE2MCIgeT0iODUiIHdpZHRoPSI0MDAiIGhlaWdodD0iMTgwIiBmaWxsPSIjZjFmNWY5
IiByeD0iNCIvPgoKICA8IS0tIFRpbWVsaW5lIGdyaWRsaW5lcyAtLT4KICA8bGluZSB4MT0iMjYwI
iB5MT0iODUiIHgyPSIyNjAiIHkyPSIyNjUiIHN0cm9rZT0iI2UyZThmMCIgc3Ryb2tlLXdpZHRoPS
IxIi8+CiAgPGxpbmUgeDE9IjM2MCIgeTE9Ijg1IiB4Mj0iMzYwIiB5Mj0iMjY1IiBzdHJva2U9IiN
lMmU4ZjAiIHN0cm9rZS13aWR0aD0iMSIvPgogIDxsaW5lIHgxPSI0NjAiIHkxPSI4NSIgeDI9IjQ2
MCIgeTI9IjI2NSIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva2Utd2lkdGg9IjEiLz4KCiAgPCEtLSBTZ
XJ2aWNlIExhYmVscyAtLT4KICA8dGV4dCB4PSIxNTAiIHk9IjExNSIgZm9udC1mYW1pbHk9IkFyaW
FsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0iZW5
kIj5Gcm9udGVuZDwvdGV4dD4KICA8dGV4dCB4PSIxNTAiIHk9IjE1NSIgZm9udC1mYW1pbHk9IkFy
aWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0iZ
W5kIj5DaGVja291dDwvdGV4dD4KICA8dGV4dCB4PSIxNTAiIHk9IjE5NSIgZm9udC1mYW1pbHk9Ik
FyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0
iZW5kIj5QYXltZW50PC90ZXh0PgogIDx0ZXh0IHg9IjE1MCIgeT0iMjM1IiBmb250LWZhbWlseT0i
QXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiMzMzMiIHRleHQtYW5jaG9yP
SJlbmQiPk5vdGlmaWNhdGlvbjwvdGV4dD4KCiAgPCEtLSBGcm9udGVuZCBzcGFuICg1MG1zID0gMT
AlID0gNDBweCkgLS0+CiAgPHJlY3QgeD0iMTYwIiB5PSIxMDAiIHdpZHRoPSI0MCIgaGVpZ2h0PSI
yNCIgcng9IjQiIGZpbGw9InVybCgjZmFzdEdyYWQpIiBmaWx0ZXI9InVybCgjY3BTaGFkb3cpIi8+
CiAgPHRleHQgeD0iMTgwIiB5PSIxMTYiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZ
m9udC1zaXplPSIxMCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjUwbXM8L3RleH
Q+CiAgPHRleHQgeD0iMjIwIiB5PSIxMTYiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiI
gZm9udC1zaXplPSIxMCIgZmlsbD0iIzIyYzU1ZSI+KDEwJSk8L3RleHQ+CgogIDwhLS0gQ2hlY2tv
dXQgc3BhbiAoMTAwbXMgPSAyMCUgPSA4MHB4KSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjE0MCIgd
2lkdGg9IjgwIiBoZWlnaHQ9IjI0IiByeD0iNCIgZmlsbD0idXJsKCNtZWRHcmFkKSIgZmlsdGVyPS
J1cmwoI2NwU2hhZG93KSIvPgogIDx0ZXh0IHg9IjI0MCIgeT0iMTU2IiBmb250LWZhbWlseT0iQXJ
pYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0i
bWlkZGxlIj4xMDBtczwvdGV4dD4KICA8dGV4dCB4PSIzMDAiIHk9IjE1NiIgZm9udC1mYW1pbHk9I
kFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjZjU5ZTBiIj4oMjAlKTwvdG
V4dD4KCiAgPCEtLSBQYXltZW50IHNwYW4gKDMwMG1zID0gNjAlID0gMjQwcHgpIC0gQ1JJVElDQUw
gUEFUSCAtLT4KICA8cmVjdCB4PSIyODAiIHk9IjE4MCIgd2lkdGg9IjI0MCIgaGVpZ2h0PSIyNCIg
cng9IjQiIGZpbGw9InVybCgjc2xvd0dyYWQpIiBmaWx0ZXI9InVybCgjY3BTaGFkb3cpIi8+CiAgP
HRleHQgeD0iNDAwIiB5PSIxOTYiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC
1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWl
kZGxlIj4zMDBtcyAoNjAlKTwvdGV4dD4KCiAgPCEtLSBDcml0aWNhbCBwYXRoIGluZGljYXRvciAt
LT4KICA8cmVjdCB4PSIyNzUiIHk9IjE3NiIgd2lkdGg9IjI1MCIgaGVpZ2h0PSIzMiIgcng9IjYiI
GZpbGw9Im5vbmUiIHN0cm9rZT0iI2VmNDQ0NCIgc3Ryb2tlLXdpZHRoPSIzIiBzdHJva2UtZGFzaG
FycmF5PSI1LDMiLz4KICA8dGV4dCB4PSI0MDAiIHk9IjIyMCIgZm9udC1mYW1pbHk9IkFyaWFsLCB
zYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2VmNDQ0
NCIgdGV4dC1hbmNob3I9Im1pZGRsZSI+Q1JJVElDQUwgUEFUSCAtIEJvdHRsZW5lY2s8L3RleHQ+C
gogIDwhLS0gTm90aWZpY2F0aW9uIHNwYW4gKDUwbXMgPSAxMCUgPSA0MHB4KSAtLT4KICA8cmVjdC
B4PSI1MjAiIHk9IjIyOCIgd2lkdGg9IjQwIiBoZWlnaHQ9IjI0IiByeD0iNCIgZmlsbD0idXJsKCN
mYXN0R3JhZCkiIGZpbHRlcj0idXJsKCNjcFNoYWRvdykiLz4KICA8dGV4dCB4PSI1NDAiIHk9IjI0
NCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJ3a
```

~~~
Gl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+NTBtczwvdGV4dD4KICA8dGV4dCB4PSI1NzUiIHk9Ij
I0NCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSI
jMjJjNTVlIj4oMTAlKTwvdGV4dD4KCiAgPCEtLSBMZWdlbmQgLyBJbnNpZ2h0IGJveCAtLT4KICA8
cmVjdCB4PSI1ODAiIHk9Ijg1IiB3aWR0aD0iMjAwIiBoZWlnaHQ9IjExMCIgcng9IjYiIGZpbGw9I
iNmZmYiIHN0cm9rZT0iI2UyZThmMCIgc3Ryb2tlLXdpZHRoPSIyIi8+CiAgPHRleHQgeD0iNjgwIi
B5PSIxMDUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9
udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiMzMzMiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk9wdGltaXph
dGlvbiBJbXBhY3Q8L3RleHQ+CgogIDxyZWN0IHg9IjU5NSIgeT0iMTE1IiB3aWR0aD0iMTIiIGhla
WdodD0iMTIiIHJ4PSIyIiBmaWxsPSJ1cmwoI3Nsb3dHcmFkKSIvPgogIDx0ZXh0IHg9IjYxNSIgeT
0iMTI1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw
9IiMzMzMiPlBheW1lbnQ6IDYwJTwvdGV4dD4KICA8dGV4dCB4PSI2MTUiIHk9IjEzOCIgZm9udC1m
YW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIj5Pc
HRpbWl6ZSBmaXJzdCE8L3RleHQ+CgogIDxyZWN0IHg9IjU5NSIgeT0iMTUwIiB3aWR0aD0iMTIiIG
hlaWdodD0iMTIiIHJ4PSIyIiBmaWxsPSJ1cmwoI21lZEdyYWQpIi8+CiAgPHRleHQgeD0iNjE1IiB
5PSIxNjAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmls
bD0iIzMzMyI+Q2hlY2tvdXQ6IDIwJTwvdGV4dD4KCiAgPHJlY3QgeD0iNTk1IiB5PSIxNzUiIHdpZ
HRoPSIxMiIgaGVpZ2h0PSIxMiIgcng9IjIiIGZpbGw9InVybCgjZmFzdEdyYWQpIi8+CiAgPHRleH
QgeD0iNjE1IiB5PSIxODUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXp
lPSIxMCIgZmlsbD0iIzMzMyI+T3RoZXJzOiAyMCU8L3RleHQ+CgogIDwhLS0gUXVlcnkgU2VjdGlv
biAtLT4KICA8cmVjdCB4PSI1ODAiIHk9IjIwNSIgd2lkdGg9IjIwMCIgaGVpZ2h0PSI4NSIgcng9I
jYiIGZpbGw9IiMxZTI5M2IiLz4KICA8dGV4dCB4PSI2ODAiIHk9IjIyNSIgZm9udC1mYW1pbHk9Ik
FyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0
iIzk0YTNiOCIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RmluZCBDcml0aWNhbCBQYXRoPC90ZXh0Pgog
IDx0ZXh0IHg9IjU5NSIgeT0iMjQ1IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9I
jEwIiBmaWxsPSIjMjJjNTVlIj5mZXRjaDwvdGV4dD4KICA8dGV4dCB4PSI2MjUiIHk9IjI0NSIgZm
9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2Y4ZmFmYyI+c3BhbnM
8L3RleHQ+CiAgPHRleHQgeD0iNTk1IiB5PSIyNjAiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZv
bnQtc2l6ZT0iMTAiIGZpbGw9IiMxNDk2ZmYiPnwgc3VtbWFyaXplPC90ZXh0PgogIDx0ZXh0IHg9I
jU5NSIgeT0iMjc1IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPS
IjZjhmYWZjIj4gIHRvdGFsX21zID0gc3VtKGR1cmF0aW9uKTwvdGV4dD4KICA8dGV4dCB4PSI1OTU
iIHk9IjI5MCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzE0
OTZmZiI+ICBieTo8L3RleHQ+CiAgPHRleHQgeD0iNjIwIiB5PSIyOTAiIGZvbnQtZmFtaWx5PSJtb
25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiNmYmJmMjQiPntzZXJ2aWNlLm5hbWV9PC90ZX
h0PgoKICA8IS0tIEJvdHRvbSBpbnNpZ2h0IC0tPgogIDxyZWN0IHg9IjMwIiB5PSIyOTUiIHdpZHR
oPSI1MzAiIGhlaWdodD0iMzUiIHJ4PSI2IiBmaWxsPSIjZmVmM2M3IiBzdHJva2U9IiNmYmJmMjQi
IHN0cm9rZS13aWR0aD0iMSIvPgogIDx0ZXh0IHg9IjUwIiB5PSIzMTgiIGZvbnQtZmFtaWx5PSJBc
mlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsbD0iIzkyNDAwZSI+T3B0aW1pemluZy
B0aGUgUGF5bWVudCBzZXJ2aWNlICgzMDBtcykgd291bGQgcmVkdWNlIHRyYWNlIHRpbWUgYnkgdXA
gdG8gNjAlLiBGb2N1cyBvcHRpbWl6YXRpb24gZWZmb3J0cyBoZXJlIGZpcnN0LjwvdGV4dD4KPC9z
dmc+Cg==)
~~~

~~~~
```dql
// Find services contributing most to total trace time
fetch spans
| summarize {
    total_self_time_ms = sum(duration) / 1000000,
~~~~

~~~
    span_count = count(),
    avg_duration_ms = avg(duration) / 1000000,
    max_duration_ms = max(duration) / 1000000
  }, by: {service.name}
| sort total_self_time_ms desc
| limit 15
```

```dql
// Find slowest operations across all services
fetch spans
| filter span.kind == "server"
| summarize {
    call_count = count(),
    avg_duration_ms = avg(duration) / 1000000,
    p99_duration_ms = percentile(duration, 99) / 1000000,
    total_time_ms = sum(duration) / 1000000
  }, by: {service.name, span.name}
| filter call_count > 10
| sort p99_duration_ms desc
| limit 20
```

```dql
// Identify high-impact optimization candidates
// (high volume + high latency = most benefit from optimization)
fetch spans
| filter span.kind == "server"
| summarize {
    call_count = count(),
    avg_duration_ms = avg(duration) / 1000000,
    total_time_ms = sum(duration) / 1000000
  }, by: {service.name, span.name}
| filter call_count > 50
| fieldsAdd impact_score = call_count * avg_duration_ms
| sort impact_score desc
| limit 20
```

---

## Summary

In this notebook, you learned:

✅ **Service discovery** - Find all services and their operations
✅ **Dependency mapping** - Use CLIENT spans with `peer.service` and
`server.address`
~~~

```
✅ **Client-server patterns** - Analyze inbound/outbound call ratios
✅ **Async messaging** - Track PRODUCER/CONSUMER spans through message queues
✅ **Trace hierarchy** - Understand span parent-child relationships
✅ **Cross-service latency** - Find latency hot spots between services
✅ **Critical path analysis** - Identify bottlenecks for optimization

---

## Next Steps

Continue to **SPANS-05: Advanced Span Analytics** to learn:
- Time series analysis and trending
- Complex aggregations and calculations
- Building dashboard-ready queries
- Alerting patterns
```