opentext™ Cybersecurity

Making LDAP Observable

Modernizing jldap and tracing LDAP operations with OpenTelemetry

Norbert Klasen

Speaker

Norbert Klasen is a Principal Consultant in the Identity and Access Management Practice at OpenText Professional Services.

His identity management career began in Tübingen, where he completed his diploma thesis at the DFN Competence Center for Directory Services in 2001 and worked for DAASI International.

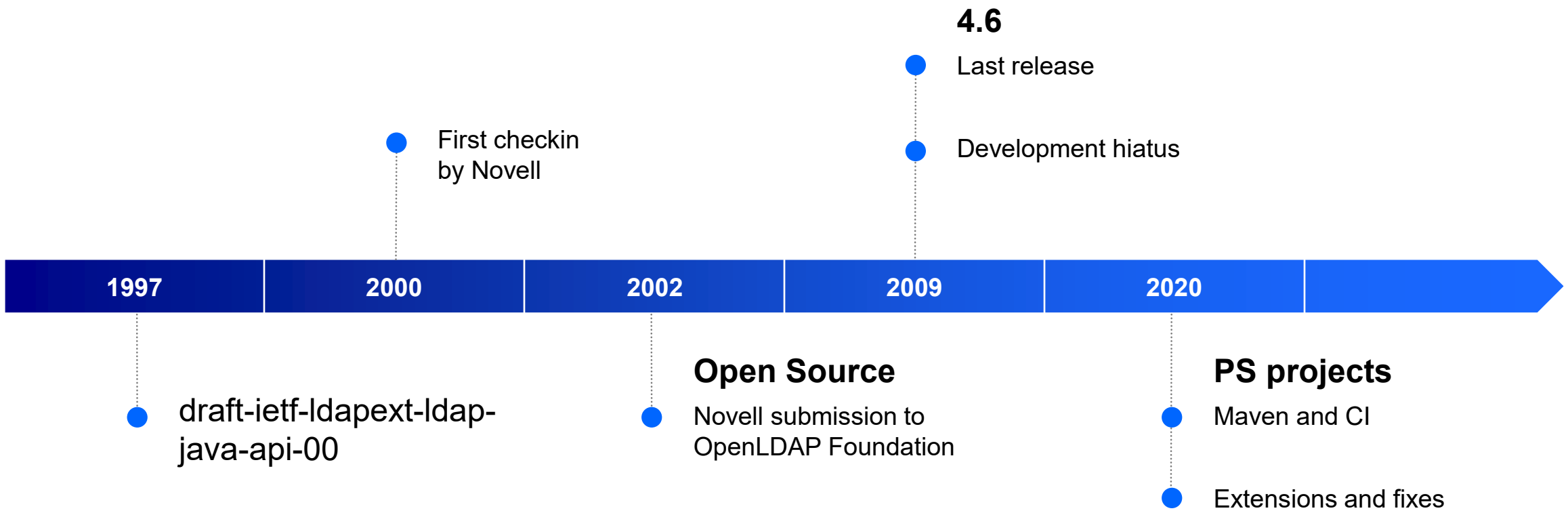
He joined Novell in 2007 and has remained with the organization through its evolution into Attachmate, Micro Focus, and OpenText.

His current focus areas include identity governance, observability, and leveraging AI to convert requirements and solution designs into low-code implementations.

OpenText ©2025. All rights reserved.

jldap

jldap timeline - LDAP Classes for Java



OpenText ©2025. All rights reserved.

NetIQ eDirectory Event System

Auditing use case for capturing events for LDAP operations

eDirectory is build with an internal event system

Modules can register for certain types of events

Events can be streamed over LDAP as IntermediateResponse messages

Implemented Java classes for these events

OpenText ©2025. All rights reserved.

Starting point and constraints

jldap v4.6 code base

Java 1.2

ant

compile time debug and release builds

Preserve API compatibility

Incremental regression-tested changes

Reliability before new features

OpenText ©2025. All rights reserved.

Chores

Reorganize sources into Maven project

Split off DSML parts into separate Maven module

Make it compile with Java 1.8

Add GitLab CI

Fix JavaDoc

OpenText ©2025. All rights reserved.

Fixes

Honor connect timeouts with LDAPJSSE factories

Synchronous TLS handshake initiation to fail fast on certificate issues

Propagate ReaderThread exceptions to calling contexts

Handle LDAP controls with absent values

Check response code before casting to LDAPExtendedResponse

Add leading asterisk when creating a string representation of a filter

Remove extra space from string representation of Attribute Type definitions

Support reading rootDSE entry from LDIF

Fix character set used in LDIFReader (ITS#6347)

Fix usage and collective attributes parsing in LDAPAttributeSchema (IST#7539)

OpenText ©2025. All rights reserved.

Enhancements

New extended operations and controls

- eDirectory Event Services
- eDirectory ServerCertificate
- NMAS Delete Password

Import samples and adapt for Java 8

Make LDAP_DEBUG flag a runtime-configurable property

Replace custom debug internals with SLF4J while preserving runtime toggles

OpenText ©2025. All rights reserved.

Roadmap

Merge a stack of pull request into jldap/main

CI to sign builds

Publish to Maven Central

Cleanup ITS

OpenText ©2025. All rights reserved.

Observability

What is Observability?

Monitoring

- Answers known questions
- Based on predefined metrics and alerts
- Example: "LDAP response time exceeded 1 second"

Observability

- Answers unknown questions
- Enables root cause analysis of new failures
- Correlates telemetry across components

Three Pillars

- Logs - Detailed events
- Metrics - Quantitative measurements
- Traces - End-to-end request flow

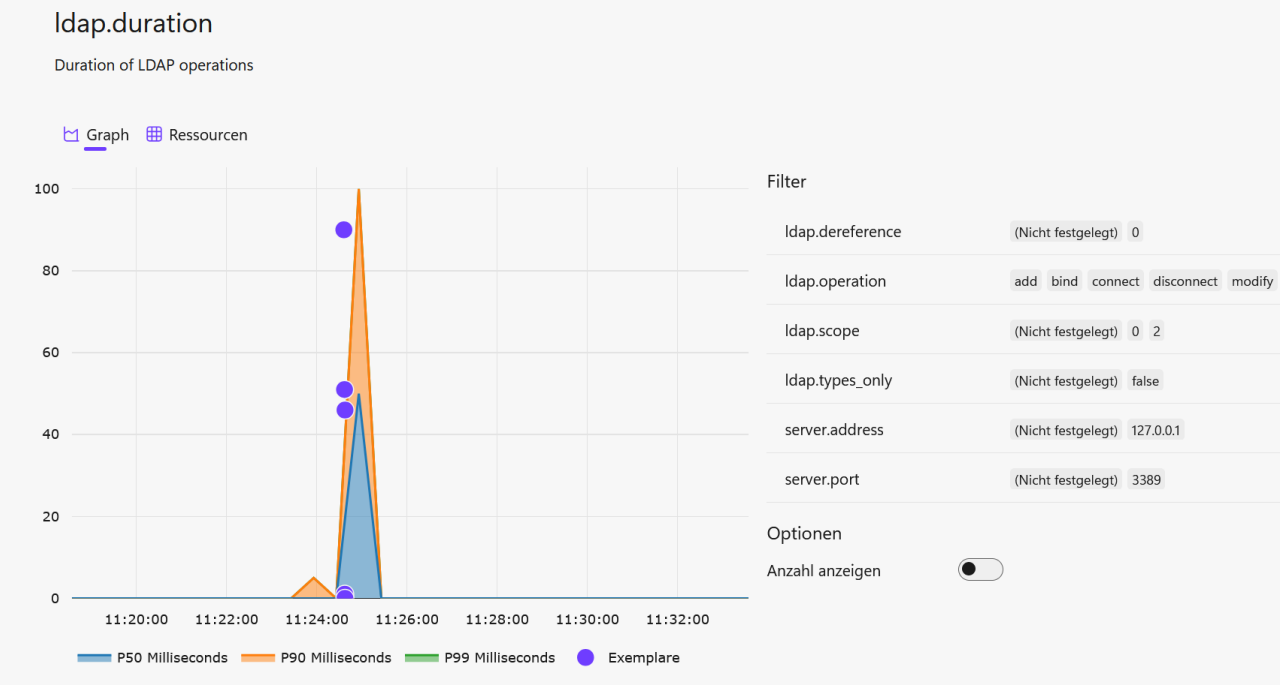
OpenTelemetry

- Vendor agnostic
- Supports many languages and frameworks
- Collects signals
- Storage and visualization is out of scope

OpenText ©2025. All rights reserved.

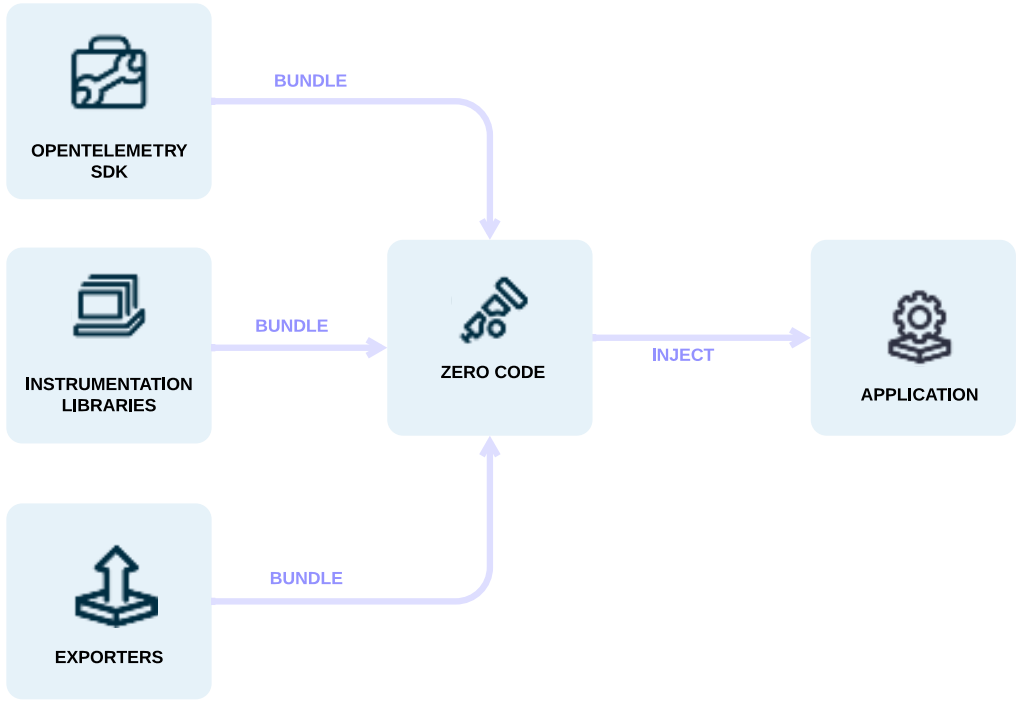
From Metrics to Traces

Metrics are usually the first step in observability. They tell you **that something is wrong**, but rarely explain **why** it is wrong. As distributed systems become more complex, teams need traces to move from symptom detection to root cause analysis.



OpenText ©2025. All rights reserved.

OpenTelemetry Zero-code instrumentation



Automatic instrumentation is available for the following languages:

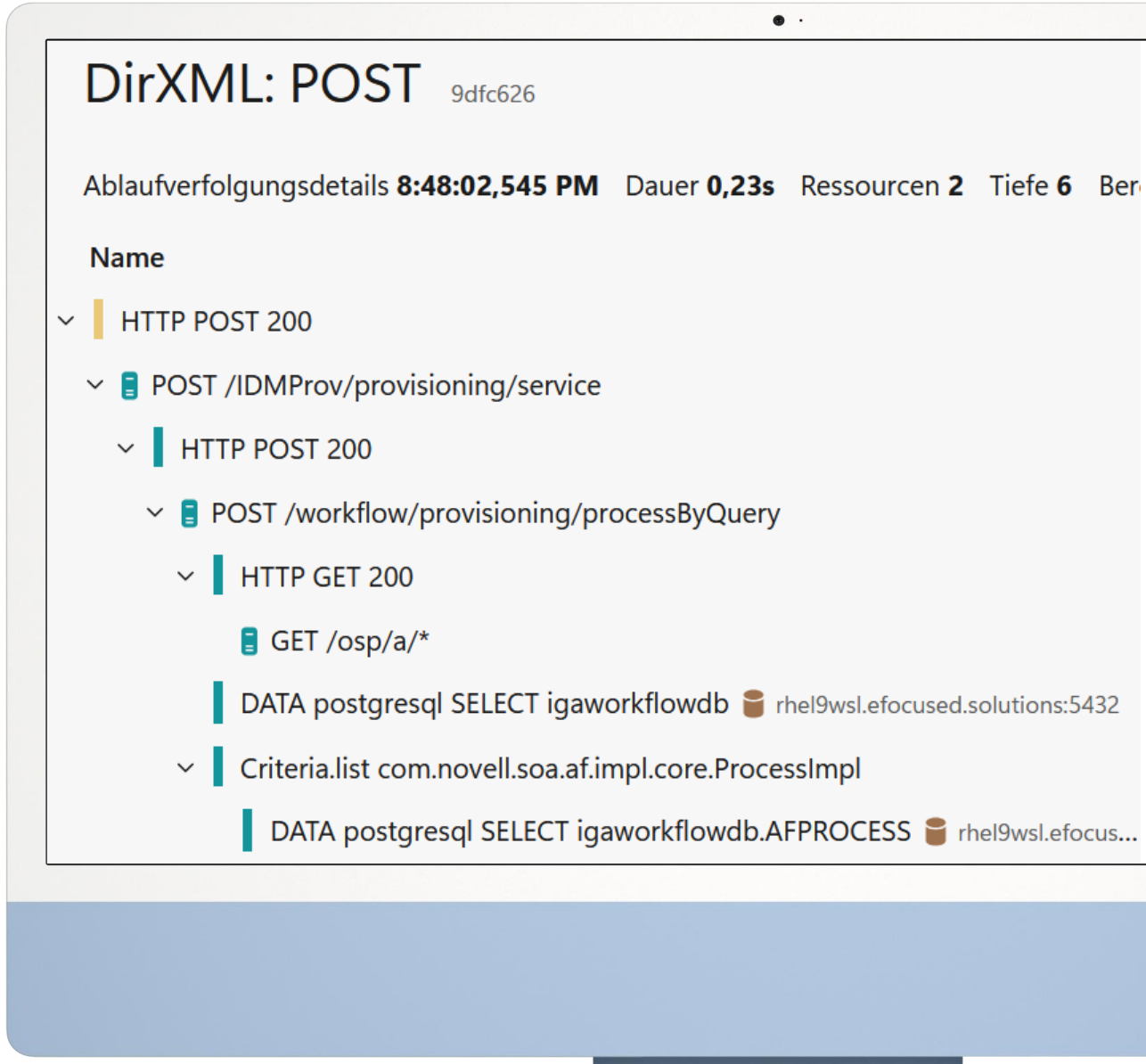
- .NET
- Go
- Java
- JavaScript
- PHP
- Python

OpenText ©2025. All rights reserved.

Many supported frameworks

In this example:

- apache-httpclient
- tomcat
- http-url-connection
- hibernate
- jdbc



I want traces to include spans for LDAP

Create Instrumentation Module for

- jldap
- JNDI

Instrument methods (all synchronous)

- connect, bind, search, read, modify, rename, add, delete

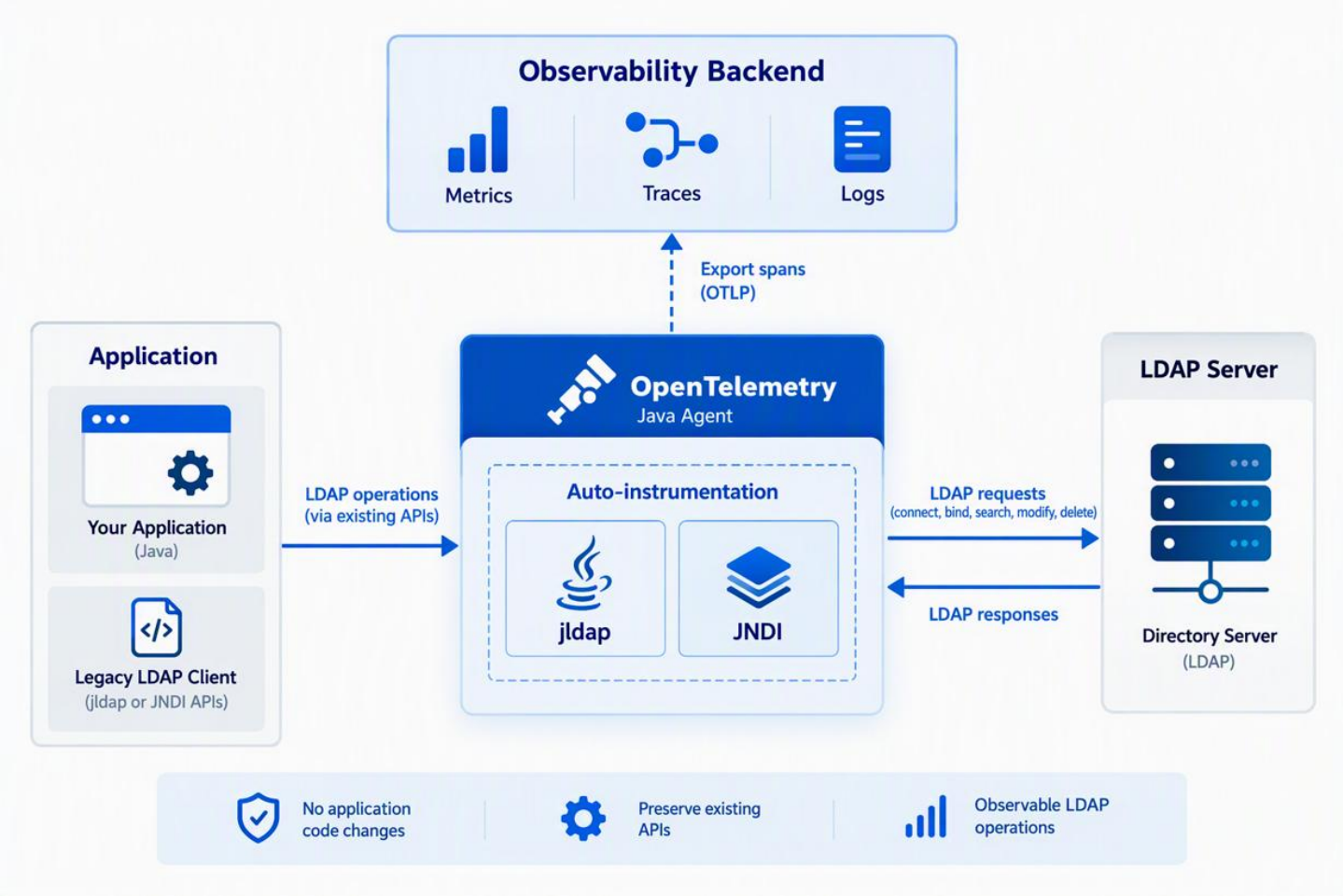
While at it, build a request duration histogram metric

Attributes: host, port, (base) dn, scope, deref

Need to be careful with sensitive/high-cardinality attributes

OpenText ©2025. All rights reserved.

Observability architecture

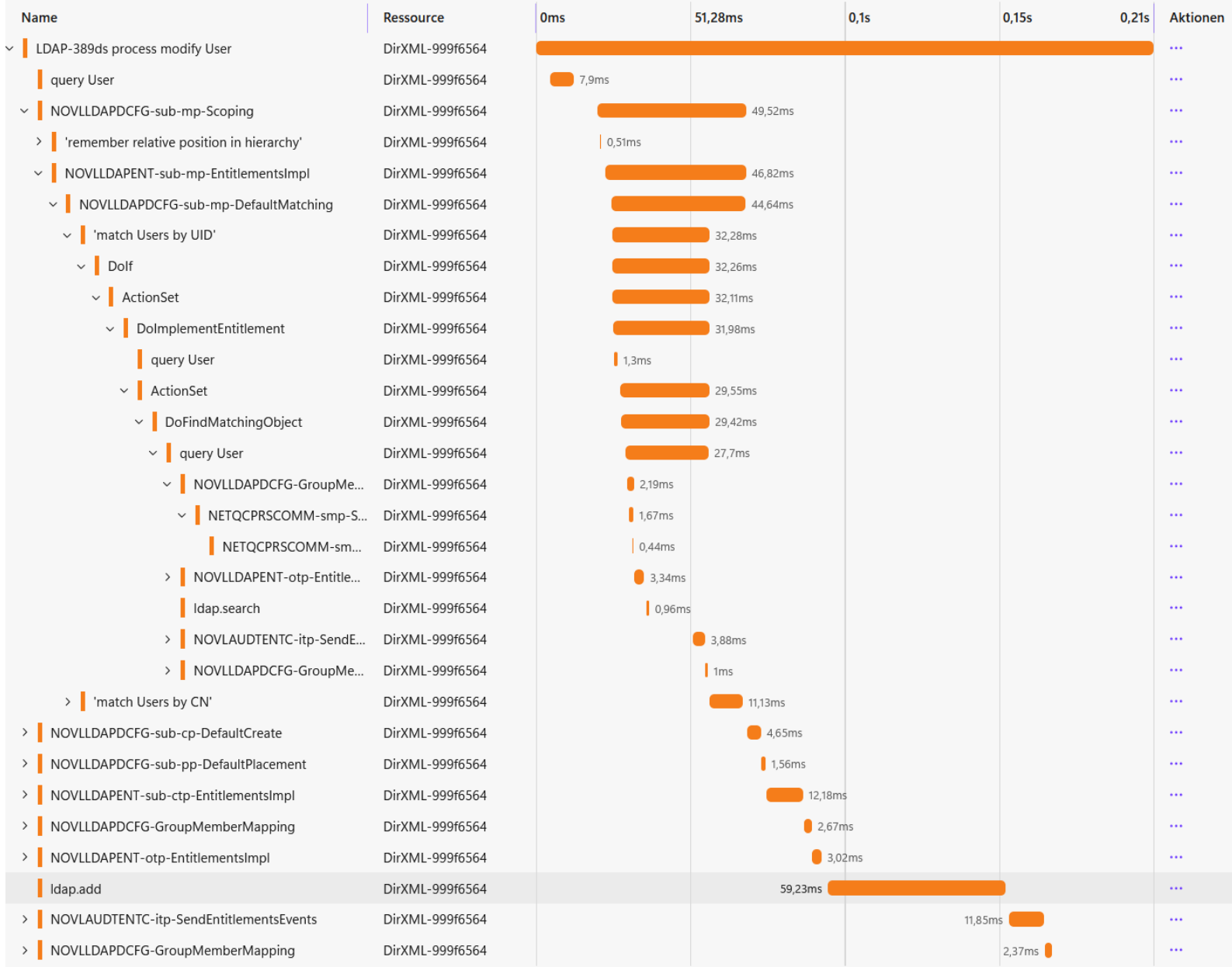


OpenText ©2025. All rights reserved.

DirXML-999f6564: LDAP-389ds process modify User b26065d

Ablaufverfolgungsdetails **9:51:30,768 PM** Dauer **0,21s** Ressourcen **1** Tiefe **15** Bereiche gesamt **110**

Filtern...



DirXML-999f6564: ldap.add 5e7b7ee

Ressource **DirXML** Dauer **59,23ms** Startzeit **96,9ms**

Span

Name	Wert
SpanId	5e7b7ee053fd0c81
Name	ldap.add
Kind	Client
instrumented.method	com.novell.ldap.LDAPConnection.add
ldap.dn	CN=achung,o=data
thread.id	52
thread.name	\WSL\system\driverset1\LDAP-389ds

Kontext

Name	Wert
Source	com.novell.ldap
TraceId	b26065d907a6ad7e2527133351db4c90
ParentId	183b87fdf3464bfc

Ressource

Name	Wert
service.name	DirXML
service.instance.id	999f6564-e65a-40c4-a26f-b7db3e7e2df9
host.arch	amd64
host.name	rhel9wsl
os.description	Linux 6.6.114.1-microsoft-standard-WSL2

Microsevices

Microsevices

Noticed the different colors?

OpenText ©2025. All rights reserved.

Distributed Tracing

The problem: A user request traverses multiple systems:

Browser -> Application -> Identity Provider -> LDAP Server -> Database

Without correlation, no common identifier exists

App Log	Request 123
LDAP Log	conn=542 op=7
Database Log	session=891

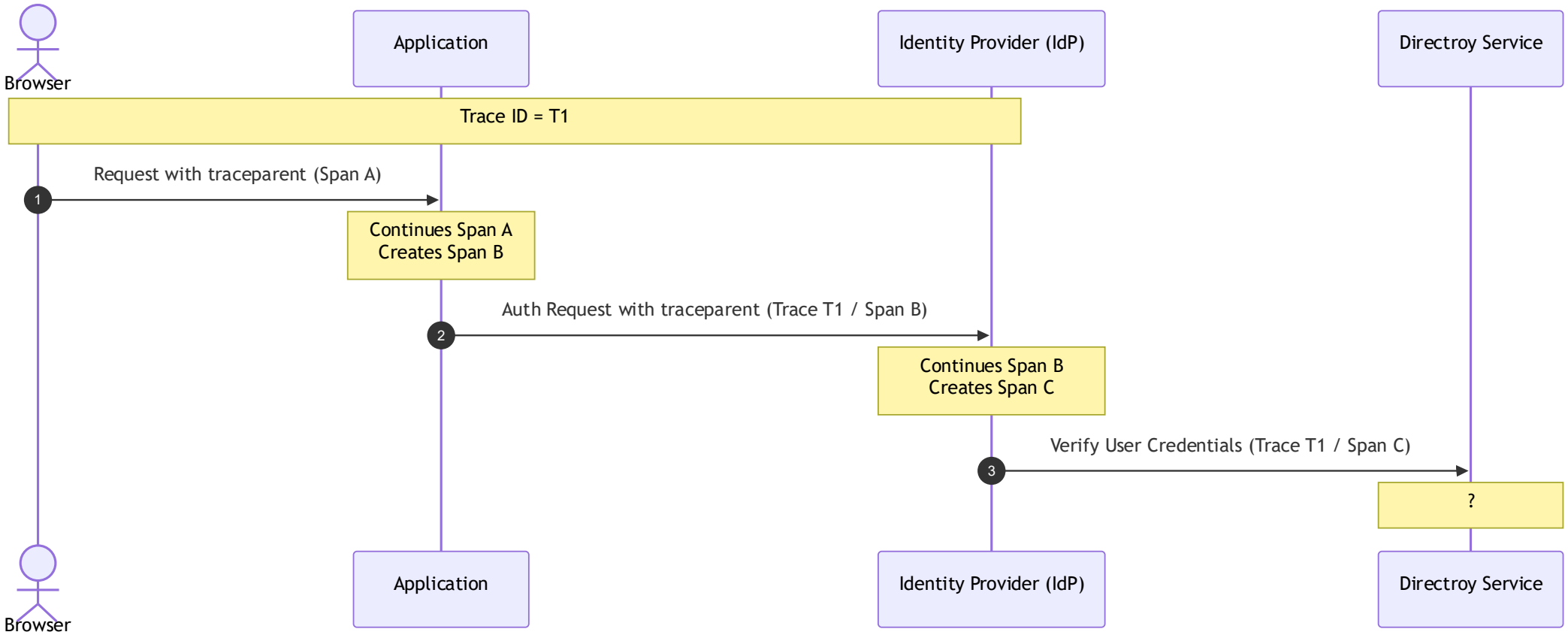
Distributed Trace

All participating services attach:

- Trace ID
- Span ID
- Parent Span ID

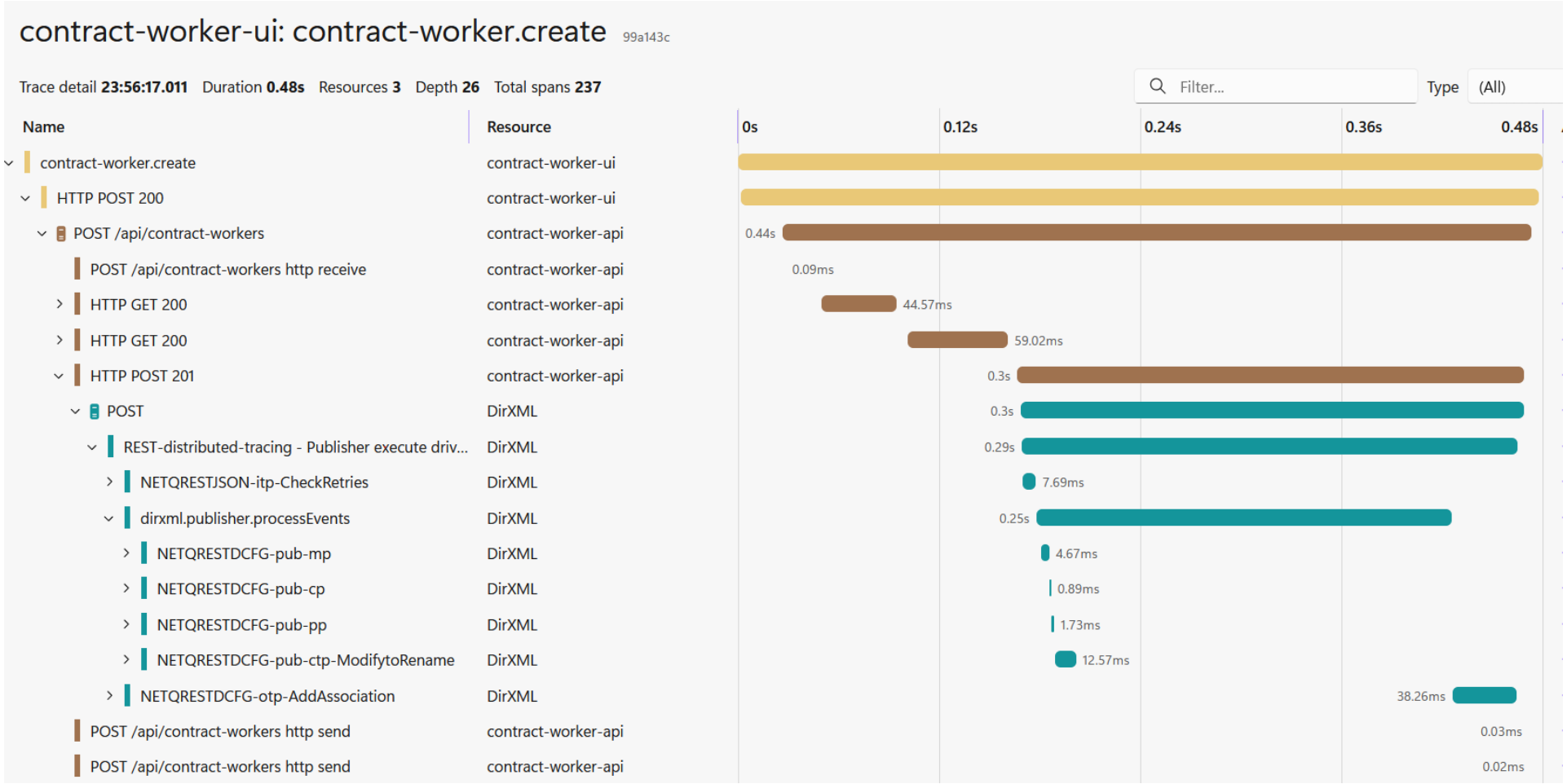
OpenText ©2025. All rights reserved.

Trace with spans



OpenText ©2025. All rights reserved.

Browser, Backend, IDM



OpenText ©2025. All rights reserved.

Trace Context Propagation

OpenTelemetry defines a **Carrier**. Fort HTTP, services propagate trace information using request **headers**. The W3C standard defines the **traceparent** header.

Example:

```
GET /login HTTP/1.1
traceparent: 00-4bf92f3577b34da6a3ce929d0e0e4736-00f067aa0ba902b7-01
```

Header Structure

- version
- trace-id
- parent-span-id
- trace-flags

Challenge: HTTP has a standard propagation mechanism. LDAP currently does not.

OpenText ©2025. All rights reserved.

Trace Context LDAP control?

Implement a Carrier

Define a new control or use existing LDAP Session Tracking Control?

```
SessionIdentifierControlValue ::= SEQUENCE {
        sessionSourceIp                 LDAPString,
        sessionSourceName               LDAPString,
        formatOID                       LDAPOID,
        sessionTrackingIdentifier       LDAPString
}
```

Few implementations exist

- 389ds (log ids)
- proftpd (client)
- OpenLDAP (proxy backends)

OpenText ©2025. All rights reserved.

LDAPSessionTrackingCarrier

Define OIDs for traceparent (and tracestate)

LDAPString works well for TextMapPropagator

Attach Session Tracking Control to LDAPConstraints of the operation

```
public static class AddAdvice {
  @Advice.OnMethodEnter(suppress = Throwable.class)
  public static Object onEnter(
      @Advice.This LDAPConnection connection,
      @Advice.Argument(0) LDAPEntry entry,
      @Advice.Argument(value = 1, readOnly = false) LDAPConstraints constraints {
    constraints =
        LDAPSessionTrackingCarrier.withTraceContext(
            constraints == null ? connection.getConstraints() : constraints);
```

OpenText ©2025. All rights reserved.

Results

389ds

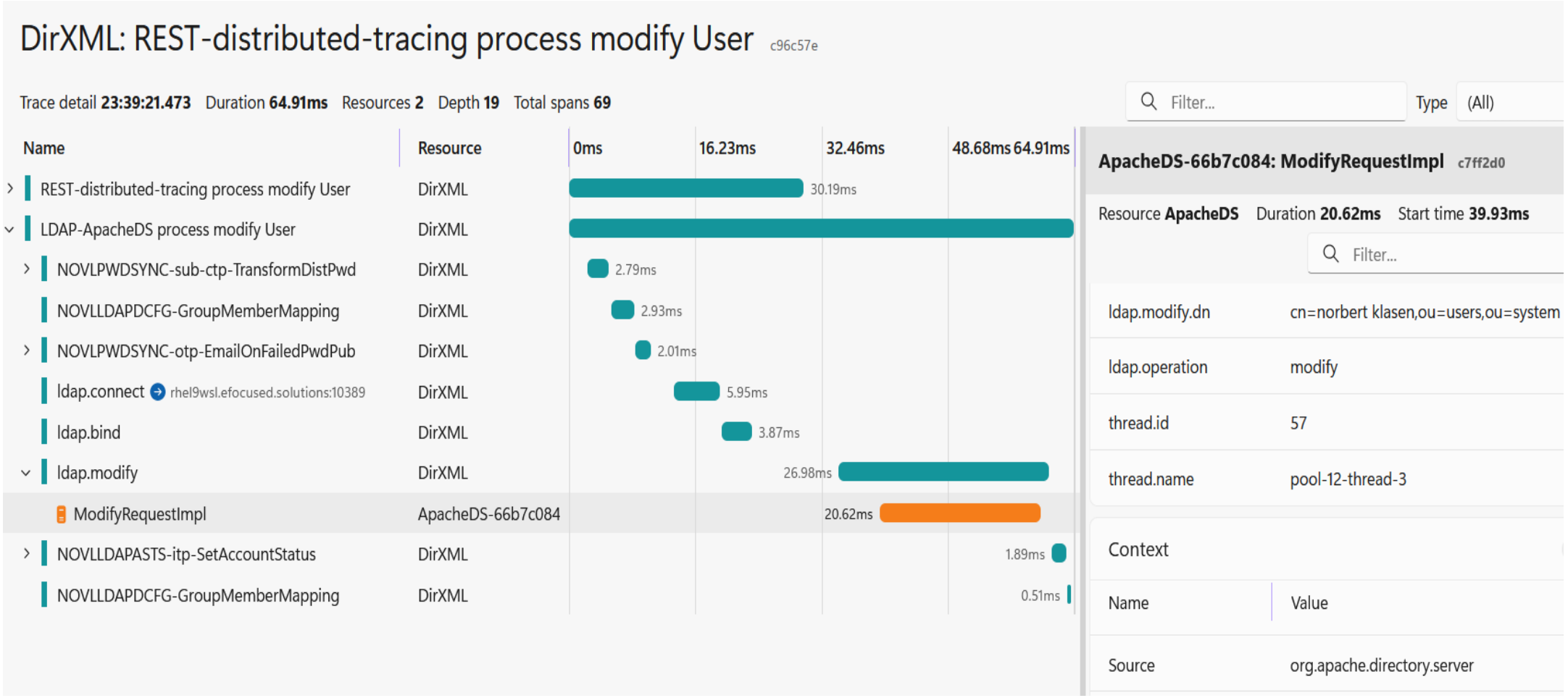
```
conn=8553 op=4 SRCH base="uid=achung,o=data" scope=0 filter="(|(objectClass=inetOrgPerson))"
attrs="objectClass"
conn=8553 op=4 RESULT err=32 tag=101 nentries=0 wtime=0.000311933 optime=0.010249199 etime=0.010559826
sid="00-b26065d907a6..."
conn=8553 op=6 ADD dn="CN=achung,o=data"
conn=8553 op=6 RESULT err=0 tag=105 nentries=0 wtime=0.034819687 optime=0.057022859 etime=0.091841355
sid="00-b26065d907a6..."
```

eDirectory

```
LDAP: (172.21.0.4:41122)(0x0002:0x63) Search request:
        base: "o=data"
        scope:2  dereference:0  sizelimit:0  timelimit:0  attrsonly:0
        filter:
"(&(|(objectClass=inetOrgPerson))(|(mail=*jack*)(givenName=*jack*)(sn=*jack*)(telephoneNumber=*jack*))
)"
LDAP: (172.21.0.4:41122)(0x0002:0x63) nds_back_search: Search Control OID 2.16.840.1.113730.3.4.18
LDAP: (172.21.0.4:41122)(0x0002:0x63) nds_back_search: Search Control OID 1.3.6.1.4.1.21008.108.63.1
```

OpenText ©2025. All rights reserved.

Context Propagation from NetIQ IDM to ApacheDS



OpenText ©2025. All rights reserved.

Community take on Trace-context propagation over LDAP

Is it worthwhile?

Would you implement it in your project?

- Client: Inject a carrier
- Server: Extract trace context from carrier

Define a new control or use Session Tracking?

Format OIDs and their semantics

Agree on semantic conventions for metrics

OpenText ©2025. All rights reserved.

opentext™

Cybersecurity

opentext.com · twitter.com/opentext · linkedin.com/company/opentext