



Microsoft Fabric

This presentation is the property of Microsoft and is intended for informational and educational purposes only. You may use, copy, and distribute this presentation for your personal, non-commercial purposes. You may not modify, alter, or create derivative works from this presentation without the prior written consent of Microsoft. You may not use this presentation to misrepresent, defame, or disparage Microsoft or its products, services, or affiliates. You may not use this presentation to endorse or promote any other products, services, or organizations without the prior written consent of Microsoft.

By using this presentation, you agree to abide by these terms. If you do not agree, you must not use this presentation. Microsoft reserves the right to change these terms and conditions at any time without notice. Microsoft disclaims any and all warranties, express or implied, relating to this presentation, including but not limited to the accuracy, completeness, timeliness, or suitability of the information contained herein. Microsoft is not liable for any damages, losses, or liabilities arising from your use of or reliance on this presentation.

Please review the terms of use posted in the content library.

The AI-native Data Engineer

#FABCONSQLCON2026

FABCON
Microsoft Fabric
COMMUNITY CONFERENCE

SQLCON
Microsoft SQL
COMMUNITY CONFERENCE

ATLANTA MARCH 16 - 20, 2026

Deepa Rajagopal
Piero Morano

Senior Product Manager
Principal PM Manager



- I lead the product work on Fabric Notebooks, including the VS Code Data Engineering extension, environments, and library management.
- Fun facts:
 - Born and raised in Italy, lived in China for 15 years and now in US.
 - Love for travels, soccer and skiing

Piero Morano: Principal PM Manager
Data Engineering Developer Experience



Deepa Rajagopal: Product Manager
Data Engineering Developer Experience

Microsoft history:

Recent: Fabric for 5
years

Prior: Bing Ads

Outside of work:

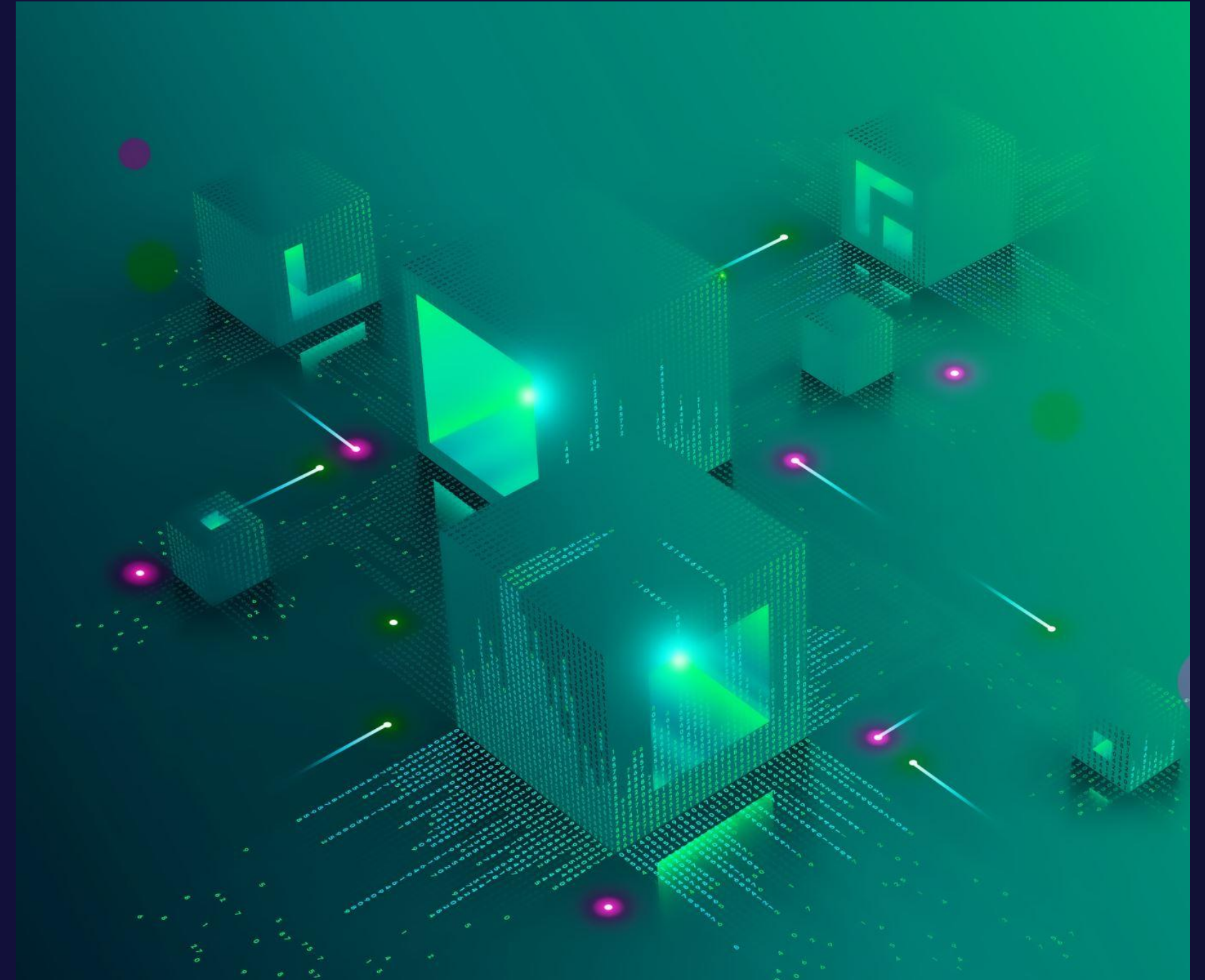
Travel with family

Cross-fit training

Creative writing

Agenda

- Introduction
- The Data Engineer (r)evolution
- The native IDE data engineer
- Web first data engineer and data analyst
- Autonomous coding
- Closing



The Data Engineer (r)evolution

With AI models generating reliable code, Data Engineers are evolving into strategic roles — doing all the thinking and planning while agents execute the work.

The Traditional Data Engineer

- ✦ Writes every line of ETL code manually
- ✦ Deep expertise in specific frameworks & tools
- ✦ Spends most time on implementation details
- ✦ Debugging connection strings & syntax errors
- ✦ Value measured by lines of code shipped
- ✦ Bottleneck: execution speed of one person



The AI-Native Data Engineer

- ✦ Thinks, plans, and designs the architecture
- ✦ Delegates execution to AI agents
- ✦ Focuses on data strategy & business outcomes
- ✦ Reviews, validates, and orchestrates agent work
- ✦ Value measured by impact and decisions made
- ✦ Multiplier: one engineer, many agents

The Right Tool for the Right Job

Background, task type, and personal preference determine the best surface for every Data Engineer.



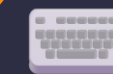
Web Platform

- ✦ Fabric portal
- ✦ Zero setup — open a browser and go
- ✦ Best for citizen developers & analysts
- ✦ Rich visual data exploration
- ✦ Collaborative editing & sharing
- ✦ Ideal for ad-hoc queries & dashboards



IDE — VS Code

- ✦ Full development environment
- ✦ GitHub Copilot deeply integrated
- ✦ Best for pro developers & data engineers
- ✦ Extensions, debugging, source control
- ✦ Multi-file editing & refactoring
- ✦ Ideal for complex pipeline development



CLI — Terminal

- ✦ Command-line agents & automation
- ✦ Copilot CLI for natural language tasks
- ✦ Best for DevOps & automation engineers
- ✦ Scriptable, repeatable, CI/CD friendly
- ✦ Headless execution — no UI needed
- ✦ Ideal for infra setup & batch operations

*There's no single "best" tool -
the right choice depends on the engineer's background, the task at
hand, and personal workflow preference.*

Explorative vs. Autonomous — Choosing the Right Mode

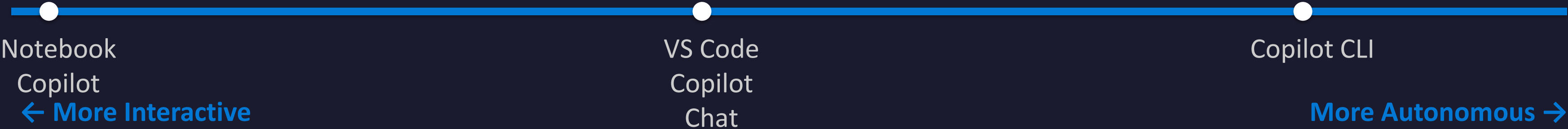
Interactive exploration needs conversation. Autonomous execution needs delegation.

Explorative — High Interaction

- ✦ Conversational back-and-forth with the AI agent
- ✦ Ideal for data profiling, debugging, ad-hoc analysis
- ✦ Best surfaces: Notebook Copilot, VS Code Copilot Chat
- ✦ The engineer guides, the agent assists in real time

Autonomous — Full Delegation

- ✦ Agent executes multi-step tasks independently
- ✦ Ideal for infrastructure setup, batch operations, CI/CD
- ✦ Best surface: Copilot CLI agents
- ✦ The engineer defines the goal, the agent delivers end-to-end



Example Scenarios

-  "Profile this dataset and find anomalies" → **Notebook Copilot**
-  "Build a transformation notebook with tests" → **VS Code + Copilot Chat**
-  "Copy all items from workspace A to B with data" → **Copilot CLI**

Live demo: VS Code Data Engineering extension overview

Scenario 1:
On-boarding an EventStream into a medallion architecture leveraging uploaded skills in builtin folder.

Scenario 1: On-boarding an EventStream

- I am reading an Eventstream using a Spark Notebook leveraging Kafka (this feature was in Amir keynote yesterday) – This is bridging Spark and RTI
- The stream comes into the Notebook as Json and I would like to transform it into a Delta Table following the best practices of Medallion Architecture and EventHub best practices
- I am not very familiar with Spark Streaming, so I rely on Copilot agent for help
- Let's see what happen

FileEditSelectionViewGoRunTerminalHelp

Workspaces [Fabric Data Engineering]

3

EXPLORER

OPEN EDITORS1 unsaved

Social_Media_Feed.ipynbPieroMSIT...

WORKSPACES [FABRIC DATA ENGINEERING]

.vscode

PieroMSIT-WS

Environments

Lakehouses

Notebooks

00_orchestration

AI_Medallion

AI_Native_Builder

Copy_LH1_to_RetailCurated

Create_Calendar_Table

DJMSITTest

Retail Lakehouse Creation

RetailAnalysis

RetailSales

Social_Media_Feed

builtin

dependencies

Social_Media_Feed.ipynb

Test4

OUTLINE

TIMELINE

Social_Media_Feed.ipynb

PieroMSIT-WS > Notebooks > Social_Media_Feed > Social_Media_Feed.ipynb > # Set the event stream item and datasource IDs

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | View Recent Runs | Outline

PySpark

Set the event stream item and datasource IDs

__in_eventstream_item_id = "0dd11d66-e91f-4da8-9a73-0a1cc47ca0b1"

__in_eventstream_datasource_id = "2249148f-3712-4def-8edb-e7801fb14d56"

Python

from pyspark.sql import SparkSession

from pyspark.sql.functions import col

from pyspark.sql.types import StringType

from pyspark.sql.dataframe import DataFrame

eventstream_options = {

"eventstream.itemid": __in_eventstream_item_id,

"eventstream.datasourceid": __in_eventstream_datasource_id

}

Read from Kafka using the config map

df_raw = spark.readStream.format("kafka").options(**eventstream_options).load()

decoded_df = df_raw.select(

col("key").cast(StringType()).alias("key"),

col("value").cast(StringType()).alias("value"),

col("partition"),

col("offset")

)

def showDf(x:DataFrame, y:int):

x.show()

Print messages to the console

query = decoded_df.writeStream.foreachBatch(showDf).outputMode("append").start()

query.awaitTermination()

Python

CLAUDE CODECHAT

RESOLVING CONFLICT IN SOCIAL MEDIA FEED NOTEBOOK

All right ready?

Social_Media_Feed.ipynb • Cell 2:13-28

Ready! Go ahead whenever you want.

Claude Opus 4.6 (1M context)(Internal only) • 6x

Social_Media_Feed.ipynb • Cell 1

An expert assistant of Fabric Data Engineering extension that helps user

FabricNotebook

Fabric Data Engineering0181Fabric Data Engineering: pimorano@microsoft.comSpaces: 4Cell 1 of 2

**Scenario 2:
Standardizing in line code in Notebooks in a
wheel package for reusability.**

Scenario 2: Standardizing in line code in Notebooks in a wheel package for reusability

- I have several Notebooks in my workspace within line logic for calculation of metrics and scores.
- I would like to package this logic in a wheel file for better reusability
- I ask Copilot for help. Copilot build the full assembly and generate the wheel package in the builtin folder
- Then I ask to modify the Notebook to remove the logic and call the package instead
- Not in the demo: upload the package to an environment for better reusability across workspaces

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

EXPLOLER

OPEN EDITORS 1 unsaved

Social_Media_Feed.ipynb PieroMSIT...

Build_RetailAnalytics_Package.ipynb...

WORKSPACES [FABRIC DATA ENGINEERING]

.vscode

PieroMSIT-WS

Environments

Lakehouses

Notebooks

00_orchestration

AI_Medallion

AI_Native_Builder

Build_RetailAnalytics_Package

builtin

dependencies

Build_RetailAnalytics_Package.i...

Copy_LH1_to_RetailCurated

Create_Calendar_Table

customer_analytics

customer_analytics Copy

DJMSITTest

inventory_sync

inventory_sync Copy

Notebook_1

ProductAnalysis

PromoUpliftSpec

reporting_pipeline

reporting_pipeline Copy

Retail Lakehouse Creation

RetailSales

RetailSalesData

RetailSalesPromotion

sales_aggregation

sales_aggregation Copy

sales_data_ingestion

sales_data_ingestion Copy

setup_variable_library

Social_Media_Feed

builtin

dependencies

Social_Media_Feed.ipynb

UpliftLogic

OUTLINE

TIMELINE

Social_Media_Feed.ipynb • Build_RetailAnalytics_Package.ipynb

PieroMSIT-WS > Notebooks > Social_Media_Feed > Social_Media_Feed.ipynb > Social Media Feed — Real-Time Data Pipeline

Generate + Code + Markdown | Run All Restart Clear All Outputs View Recent Runs Outline

PySpark

.filter(col("comment_text").isNotNull() & (trim(col("comment_text")) != ""))

)

3. Standardize, enrich, and reorder – business columns first, technical on the right

df_silver = (

df_valid

.withColumn("platform", lower(trim(col("platform"))))

.withColumn("post_type", lower(trim(col("post_type"))))

.withColumn("comment_text", trim(col("comment_text")))

.withColumn("post_timestamp", to_timestamp(col("post_datetime"), "yyyy-MM-dd HH:mm:ss"))

.withColumn("sentiment_score",

when(col("sentiment") == "Positive", 1.0)

.when(col("sentiment") == "Neutral", 0.5)

.otherwise(0.0)

)

.withColumn("post_hour", hour(to_timestamp(col("post_datetime"), "yyyy-MM-dd HH:mm:ss")))

.withColumn("post_day_of_week", dayofweek(to_timestamp(col("post_datetime"), "yyyy-MM-dd HH:mm:ss")))

.withColumn("engagement_score",

(col("likes_count").cast("double") + col("shares_count").cast("double") + col("replies_count").cast("double"))

)

.select(

Business columns first

"feedback_id",

"post_timestamp",

"platform",

"post_type",

"customer_id",

"brand_id",

"comment_text",

"sentiment",

"sentiment_score",

"rating",

"likes_count",

"shares_count",

"replies_count",

"engagement_score",

"hashtags",

"post_url",

Derived time dimensions

"post_hour",

"post_day_of_week",

Metadata / technical columns on the right

"ingestion_timestamp",

"batch_id",

"ingestion_date",

)

)

Python

CLAUDE CODE CHAT

← CREATING A .WHL PACKAGE FOR RETAIL DATA ENGINEERI... □

Restore Checkpoint

Wait

Standing by.

Claude Opus 4.6 (1M context)(Internal only) • 6x

Social_Media_Feed.ipynb • Cell 1

An expert assistant of Fabric Data Engineering extension that he

FabricNotebook Claude Opus 4.6 (1M conte... □

Local Default Approvals

Fabric Data Engineering 0 4 Fabric Data Engineering: pimirano@microsoft.com Cell 1 of 15

Scenario 3:

Bulk updates to notebooks with hardcoded values

Scenario 3: Bulk update of notebooks with hardcoded code cells

- I have several Notebooks in my workspace with hardcoded values like Lakehouse name, Database name in it.
- Instead of fixing this manually, I ask Copilot to review the notebooks and suggest improvements.
- Copilot scans through the notebooks and identifies:
 - hardcoded values and duplicated configuration.
 - that there is no mechanism to switch from dev to test to prod so this would demand manual updates.
- It suggests three simple steps:
 - Create a Variable Library with these keys
 - Update all the notebooks to reflect the changes
 - Finally validate the updated Notebooks

FileEditSelectionViewGoRunTerminalHelp

WORKSPACES [FABRIC DATA ENGINEERING]

EXPLORED

OPEN EDITORS

WORKSPACES [FABRIC DATA ENGINEERING]

sales_data_ingestion.ipynb

reporting_pipeline.ipynb

inventory_sync.ipynb

sales_aggregation.ipynb

customer_analytics.ipynb

PieroMSIT-WSPierMSIT-WS

customer_analytics

customer_analytics.ipynb

GenerateCodeMarkdownRun AllClear All OutputsView Recent RunsOutline

Customer Analytics

Segments customers by lifetime value and writes analysis results to the reporting database.

Configuration

lakehouse_name = "sales_lakehouse"

data_folder = "raw/customers/2026"

db_server = "zava-sql-prod.database.windows.net"

db_name = "SalesDB"

db_username = "analytics_user"

db_password = "Cust0m3r@2024!"

print(f"Source: {lakehouse_name}/{data_folder}")

print(f"Target DB: {db_server}/{db_name}")

Python

Read customer data from lakehouse

from pyspark.sql import Row

import random

customer_data = []

for i in range(20):

customer_data.append(Row(

customer_id=f"CUST-{i:03d}",

name=f"Customer {i}",

segment=random.choice(["Premium", "Standard", "Basic"]),

lifetime_value=random.randint(1000, 50000),

region=random.choice(["North", "South", "East", "West"])

))

customers_df = spark.createDataFrame(customer_data)

print(f"Loaded {customers_df.count()} customer records")

customers_df.show(5)

Python

Analyze customer segments

from pyspark.sql.functions import col, count, avg

segment_analysis = customers_df \

Python

CHAT

CENTRALIZING CONFIGURATION IN NOTEBOOKS WITH COPILOT

Restore Checkpoint

Ready when you are!

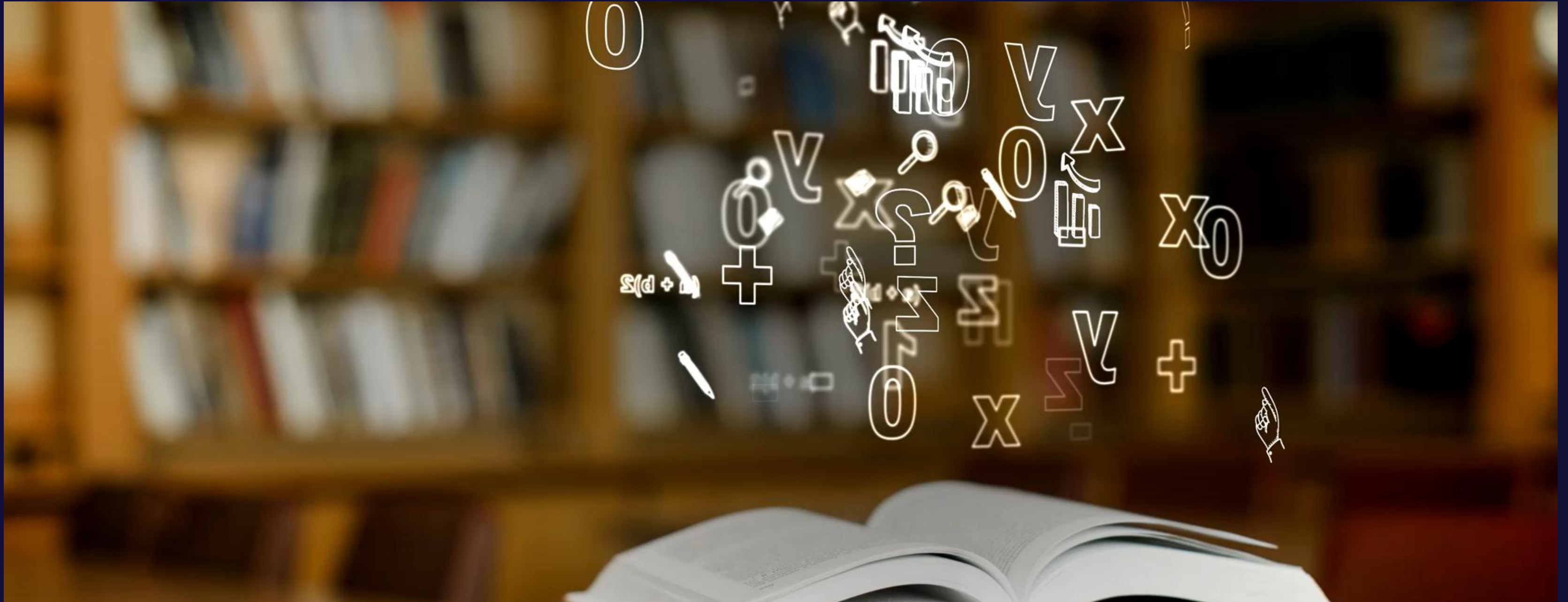
Claude Opus 4.6 (1M context)(Internal only) • 6x

Fabric Data Engineering

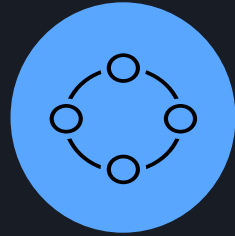
Fabric Data Engineering: pimorano@microsoft.com

Cell 1 of 5

Copilot in notebooks



What Copilot understands



Context Awareness

- Current workspace
- Notebook metadata
- Notebook cells and output
- Default lakehouse
- Runtime environment and compute



Tooling Integration

- Lakehouse data, tables, schema
- Notebook authoring & execution
- Resources interaction
- Spark monitoring & diagnostics
- Schedule management



Reasoning & Assistance

- Code analysis
- Contextual error diagnosis
- Adaptive step recommendations
- Autonomous iteration

Where Copilot helps

Explore Data

- Discover and map data models (tables, columns, relationships).
- Profile data to spot anomalies and quality issues.

Code

- Generate and optimize complex queries/transforms.
- Refactor and streamline notebook logic.
- Translate code across PySpark and Spark SQL.

Fix & Optimize

- Diagnose errors and pipeline bottlenecks.
- Recommend fixes, next analysis steps or improvements.

Scale Workflows

- Recommend next steps analysis steps or improvements.
- Organize notebooks for clarity and modular execution.
- Implement logging and manage scheduling.

Microsoft

Fabric

RetailSalesPromotion

Confidential\Microsoft Extended

Saved

Search

138

Home

Edit

AI tools

Run

View

Comments

History

Develop

Share

Run all

Standard session

PySpark (Python)

Environment

Workspace default

Data Wrangler

Copilot

Explorer

Data items

Resources

Connections

Add data items

OneLake

RetailLakehouse

1

2

3

Welcome to your new notebook

Type here in the cell editor to add code!

PySpark (Python)

Copilot

Hello Deepa!

Build autonomously with agent mode

Understand & Document

Summarize this notebook and add Markdown.

Add Markdown to explain my functions.

Show other notebooks related to this one.

Explore & Validate Data

Clean & Prepare Data

Analyze, Optimize & Convert

Ask questions and work with this notebook

Agent

AI-generated content may be incorrect

Selected Cell 1 of 1 cells

Where Copilot helps

Explore Data

- Discover and map data models (tables, columns, relationships).
- Profile data to spot anomalies and quality issues.

Code

- Generate and optimize complex queries/transforms.
- Refactor and streamline notebook logic.
- Translate code across PySpark and Spark SQL.

Diagnose & Fix

- Diagnose errors and pipeline bottlenecks.
- Recommend fixes, next analysis steps or improvements.

Scale Workflows

- Recommend next steps analysis steps or improvements.
- Organize notebooks for clarity and modular execution.
- Implement logging and manage scheduling.

Microsoft

Fabric

RetailSalesPromotion

RetailLakehouse

Confidential\Microsoft Extended

Saved

Search

Home

Workspaces

Copilot

OneLake catalog

Monitor

Real-Time

Workloads

PieroMSIT-WS

HomeEditAI toolsRunView

CommentsHistoryDevelopShare

Run allStandard sessionPySpark (Python)EnvironmentWorkspace defaultData WranglerCopilot

Explorer

Data itemsResourcesConnections

Add data items

OneLake

RetailLakehouse

+ Code+ Markdown

Copilot

Hello Deepa!

Build autonomously with agent mode

Understand & Document

Summarize this notebook and add Markdown.

Add Markdown to explain my functions.

Show other notebooks related to this one.

Explore & Validate Data

Clean & Prepare Data

Analyze, Optimize & Convert

Ask questions and work with this notebook

Agent

AI-generated content may be incorrect

Session ready

AutoSave: On

Copilot completions: On

2

Selected Cell 0 of 0 cells

Where Copilot helps

Explore Data

- Discover and map data models (tables, columns, relationships).
- Profile data to spot anomalies and quality issues.

Code

- Generate and optimize complex queries/transforms.
- Refactor and streamline notebook logic.
- Translate code across PySpark and Spark SQL.

Fix & Optimize

- Diagnose errors and pipeline bottlenecks.
- Recommend fixes, next analysis steps or improvements.

Scale Workflows

- Recommend next steps analysis steps or improvements.
- Organize notebooks for clarity and modular execution.
- Implement logging and manage scheduling.

Microsoft

Fabric

PromoUpliftSpec

Confidential\Microsoft Extended

Saved

Search

221

Home

Edit

AI tools

Run

View

Comments

History

Develop

Share

Run all

Standard session

PySpark (Python)

Environment

Workspace default

Data Wrangler

Copilot

Explorer

Data items

Resources

Connections

Add data items

OneLake

RetailLakehouse

Tables

dbo

brands

calendar

customers

inventory...

products

promotions

sales_tran...

stores

Files

Code

Markdown

Copilot

Preview

Build autonomously with agent mode

Understand & Document

Summarize this notebook and add Markdown.

Add Markdown to explain my functions.

Show other notebooks related to this one.

Explore & Validate Data

Clean & Prepare Data

Analyze, Optimize & Convert

Ask questions and work with this notebook

Agent

AI-generated content may be incorrect

Selected Cell 0 of 0 cells

Fabric

Session ready

AutoSave: On

Copilot completions: On

2

Microsoft

Fabric

UpliftLogic

ProductAnalysis

Confidential\Microsoft Extended

Search

Home

Edit

AI tools

Run

View

Comments

History

Develop

Share

Run all

Standard session

PySpark (Python)

Environment

Workspace default

Data Wrangler

Copilot

Home

Workspaces

Copilot

OneLake catalog

Monitor

Real-Time

Workloads

PieroMSIT-WS

Explorer

Data items

Resources

Connections

Add data items

OneLake

RetailLakehouse

Modular Uplift Logic Notebook

This notebook contains a reusable function and workflow for measuring promotion uplift.

Pass the new parameter `group_by_field` to group results by any valid field (e.g., `'brand_name'`, `'category'`, `'product_id'`).

To reuse in other notebooks, call this notebook and set product filters, date ranges, and grouping column as parameters.

1

PARAMETERS - Full flexibility for filtering and groupings

2

from notebookutils import mssparkutils

3

import json

4

5

def _get_param(name, default=None, paramtype=None):

6

try:

7

val = mssparkutils.runtime.get_context(name)

8

if val is None:

9

return default

10

if paramtype == list and isinstance(val, str):

11

return json.loads(val)

12

return val

13

except Exception:

14

return default

15

16

window_days = int(_get_param('window_days', 7))

17

product_category = _get_param('product_category', None)

18

promotion_type = _get_param('promotion_type', None)

19

city = _get_param('city', None)

20

region = _get_param('region', None)

21

state = _get_param('state', None)

22

year = _get_param('year', None)

23

group_by_fields = _get_param('group_by_fields', ["category"], paramtype=list)

24

--- Force convert to list if needed ---

25

if isinstance(group_by_fields, str):

26

try:

Session ready

AutoSave: On

Copilot completions: On

Selected Cell 14 of 14 cells

Microsoft

Fabric

RetailSales

Gold_Lakehouse

Confidential\Microsoft Extended

Saved

Search

Home

Edit

AI tools

Run

View

Comments

History

Develop

Share

Run all

Standard session

PySpark (Python)

Environment

Workspace default

Data Wrangler

Copilot

Home

Workspaces

Copilot

OneLake catalog

Monitor

Real-Time

Workloads

PieroMSIT-WS

...

Fabric

Explorer

Data items

Resources

Connections

Add data items

OneLake

RetailLakehouse

Tables

dbo

brands

calendar

customers

inventory_history

products

promotions

sales_transactions

stores

Files

Gold_Lakehouse

1

2

3

welcome to your new notebook

Type here in the cell editor to add code!

PySpark (Python)

Session ready

AutoSave: On

Copilot completions: On

4

Selected Cell 1 of 1 cells

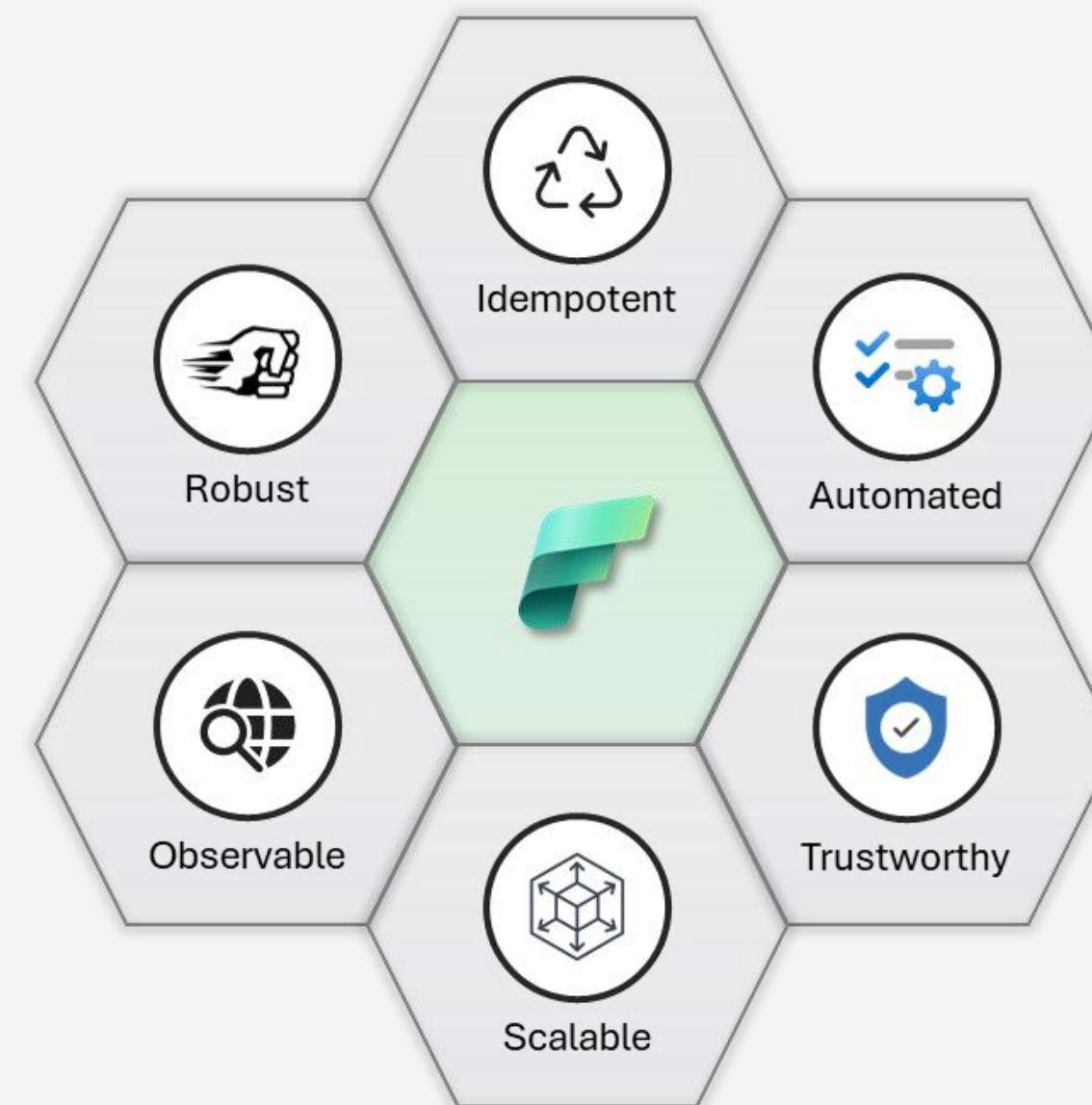
Autonomous Code:
Real life experience from a Data Engineer:
Raki Rahman



Raki Rahman

Principal Software Engineer
SQL Server Telemetry & Intelligence

SQL Telemetry & Intelligence





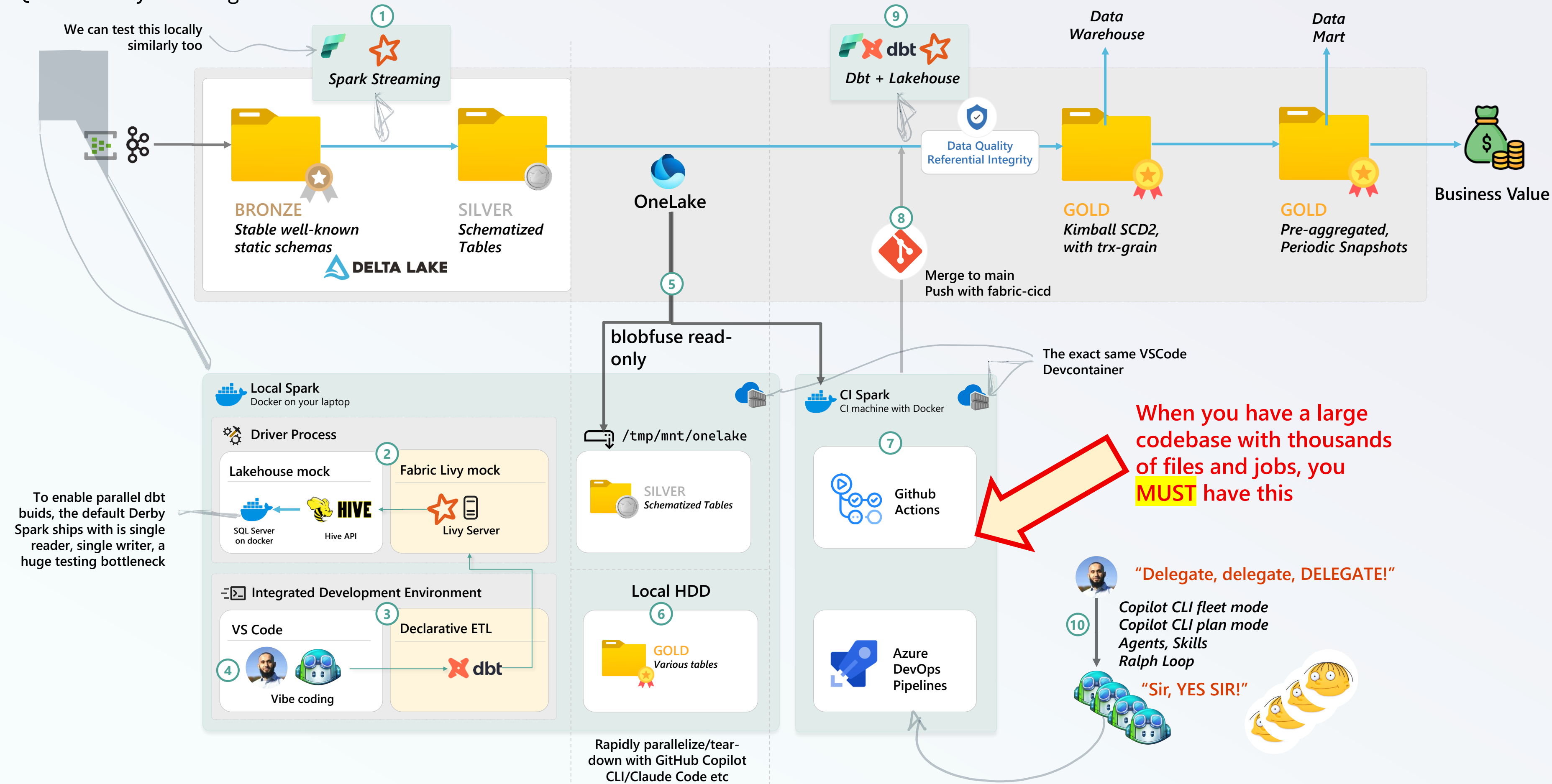
The AI-Native Data Engineer's

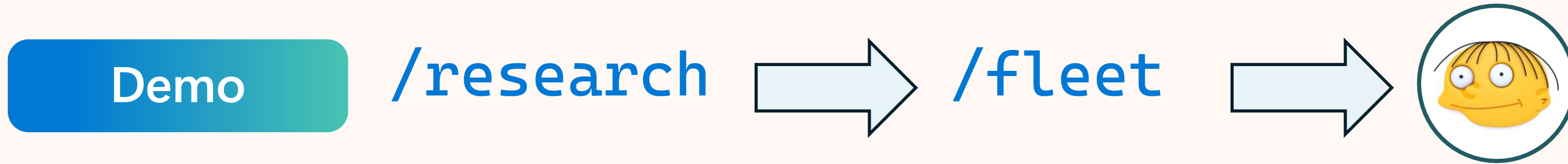
Best friend – Ralph Wiggum

See [Ralph Loop](#)

How we use Spark

SQL Telemetry & Intelligence





Autonomous development with GitHub Copilot

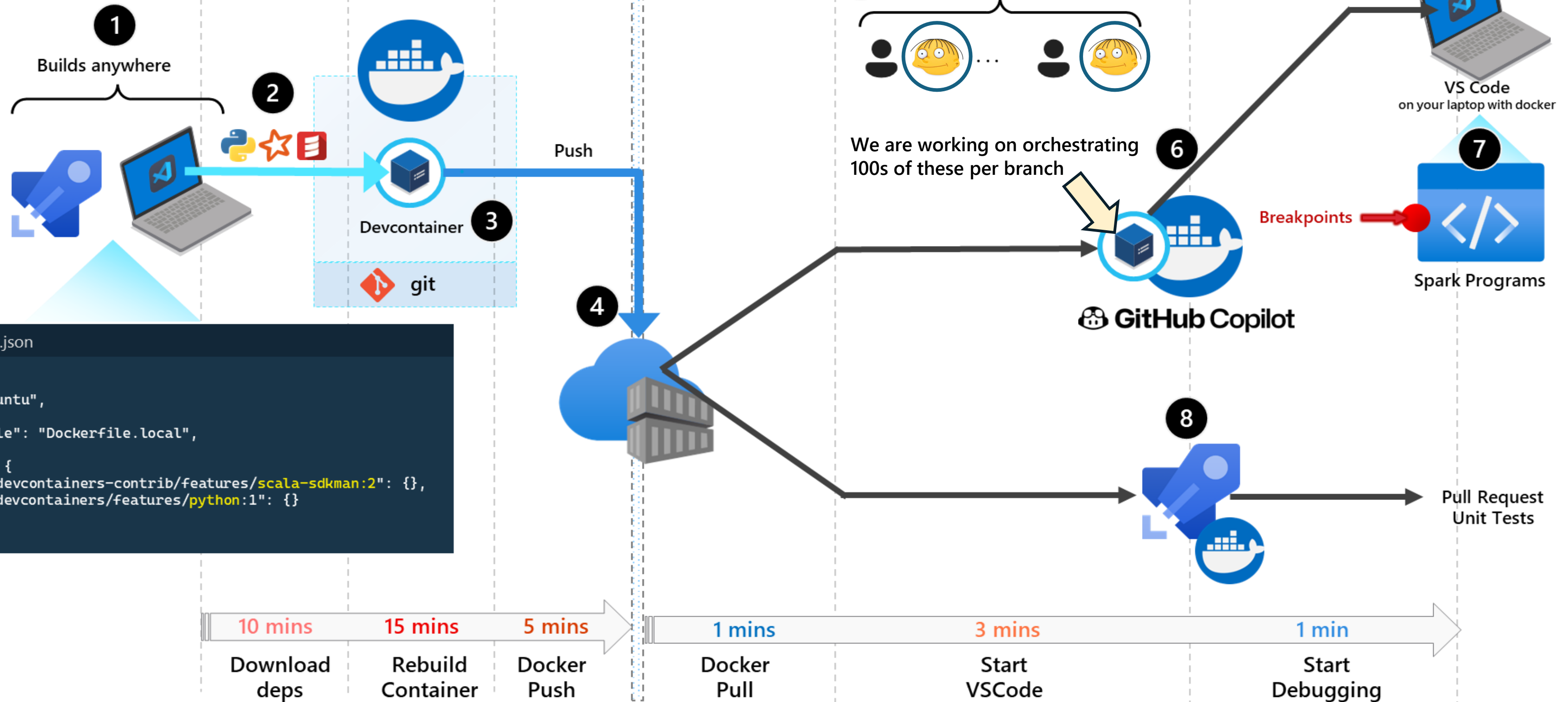
Build anywhere

Build once

Write once

Read many

Consistent



Sound off.
The mic is all yours.
Influence the product roadmap.

Join the Fabric User Panel



Share your feedback directly with our
Fabric product group and researchers.

<https://aka.ms/JoinFabricUserPanel>

Join the SQL User Panel



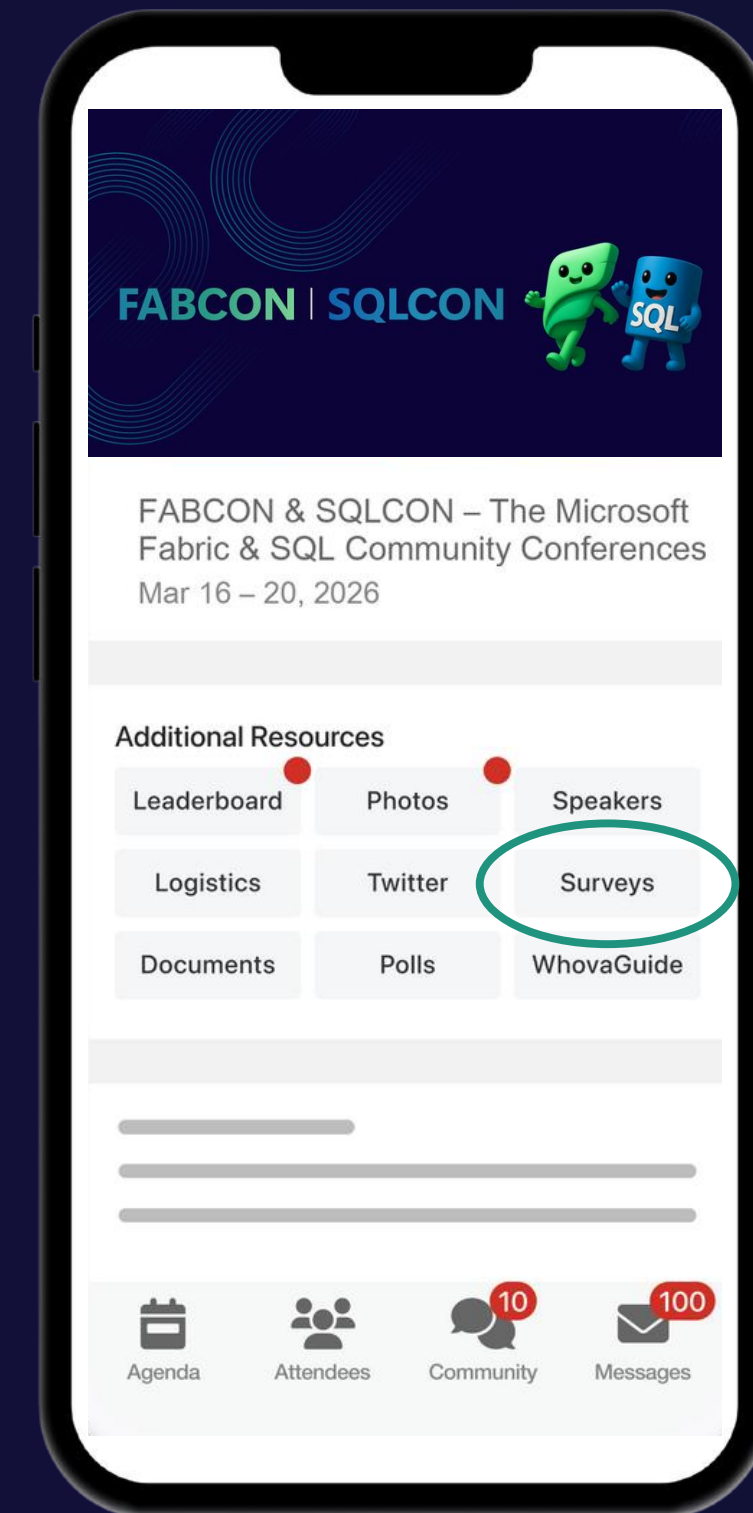
Influence our SQL roadmap and ensure
it meets your real-life needs

<https://aka.ms/JoinSQLUserPanel>

How was the session?



Complete Session Surveys in
Uhova for your chance to WIN
PRIZES!



Get Two Fabric Certifications for FREE

Attendees of FABCON can take the Fabric Analytics Engineer or Fabric Data Engineer exam for free. Be part of the 2 fastest growing role-based certifications in Microsoft history.

Request your voucher by March 23, 2026.

<https://aka.ms/fabcon/cert100>

