aspitalia.com

Exploring .NET, PostgreSQL, and Linux as your next OSS app development stack

UGIdotNET

Silvano Coriani
PostgreSQL @ Azure
in linkedin.com/in/scoriani/
github.com/scoriani

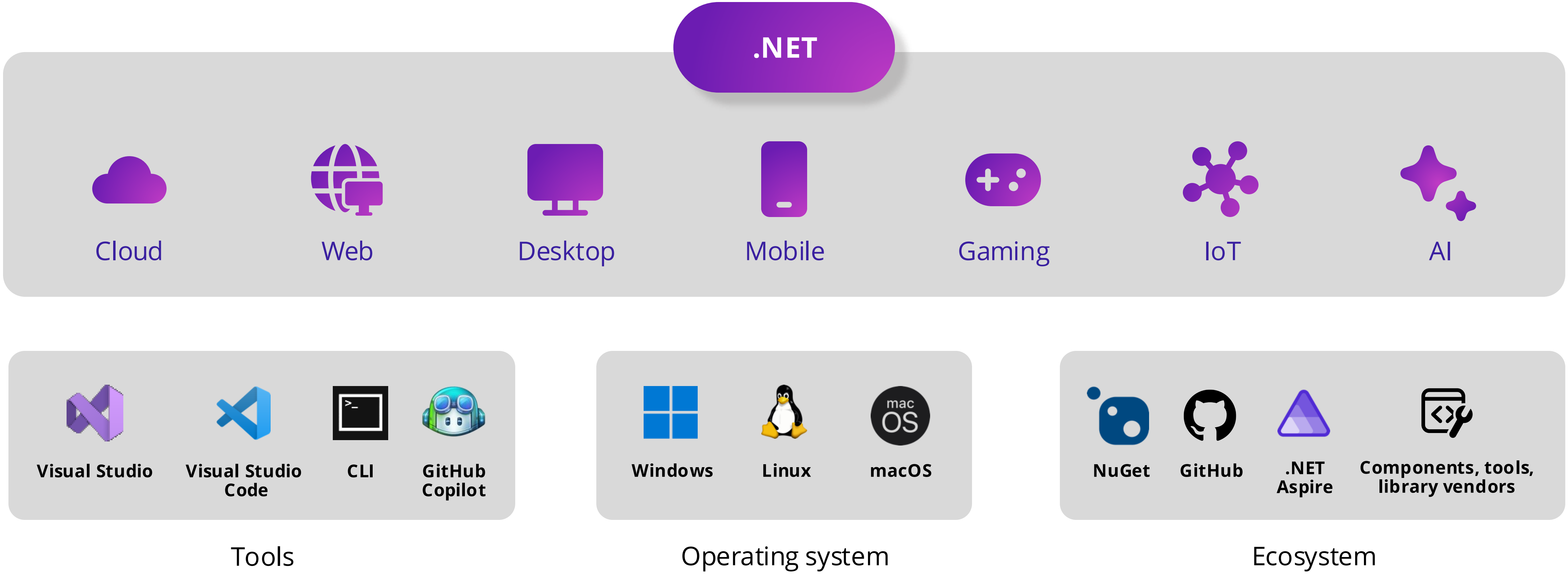
.NET Conference Italia 2024

.NET

Who am I?

- Principal Software Engineer in PostgreSQL @ Microsoft.
- Worked for ˜30 years on SQL Server in various roles ☺
- Experience in application development and database design, troubleshooting, and performance tuning.
- Specialized in data access libraries, query optimization, and distributed system design.
- Author and speaker in many international industry conferences.

Build anything with a unified platform



.NET

Cloud

Web

Desktop

Mobile

Gaming

IoT

AI

Visual Studio

Visual Studio Code

CLI

GitHub Copilot

Tools

Windows

Linux

macOS

Operating system

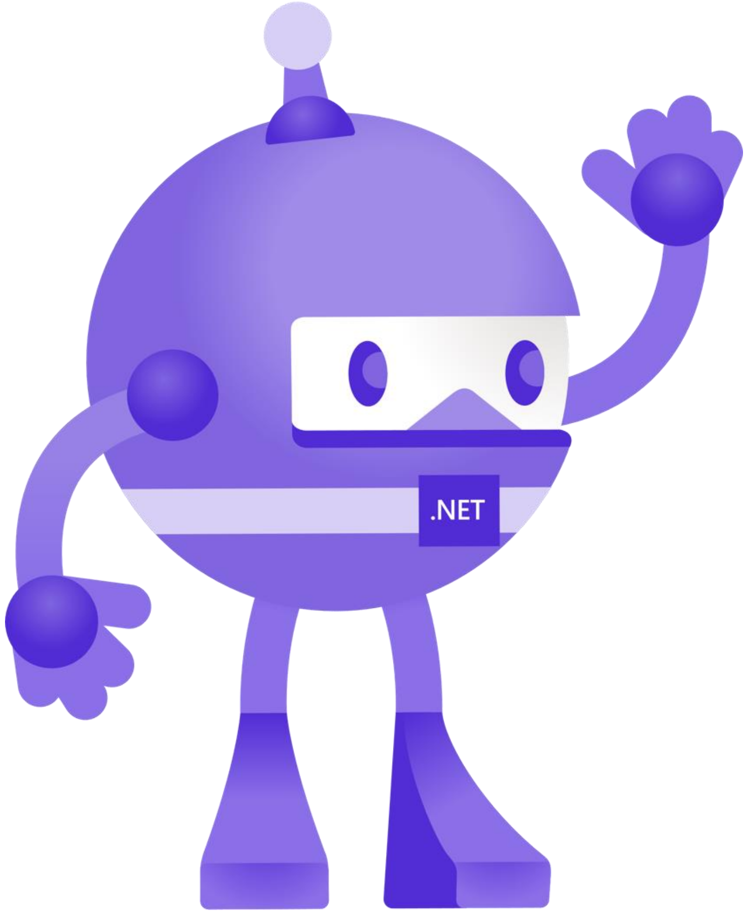
NuGet

GitHub

.NET Aspire

Components, tools, library vendors

Ecosystem


.NET

❤

Microsoft loves Linux

2009

Pigs do fly: Microsoft unleashes 20000 lines of Linux code – ZDNet

2011

- Microsoft makes the top 5 contributor list for Linux 3.0
- Hyper-V support for Linux exits the staging tree

CentOS

ubuntu

2012-2014

- Azure supports broad range of Distros
- Satya Nadella declares "Microsoft Loves Linux"
- .net Foundation Founded

2015

- VS Code Released
- Microsoft jointly forms Node.js foundation

2016-2018

- First Azure tuned Microsoft Linux kernel
- Canonical delivers Azure tuned kernels
- SUSE delivers Azure tuned kernel
- Windows Subsystem for Linux in Windows 10
- Microsoft joins Linux foundation
- GitHub recognizes Microsoft as a top open-source contributor

2018

- VS Code ranked #1 developer tool
- GitHub acquisition
- ~5,000 Microsoft employees committing to open-source projects on GitHub
- Azure trending to 50% Linux
- Microsoft continues as largest contributor to OSS on GitHub

2019-2023

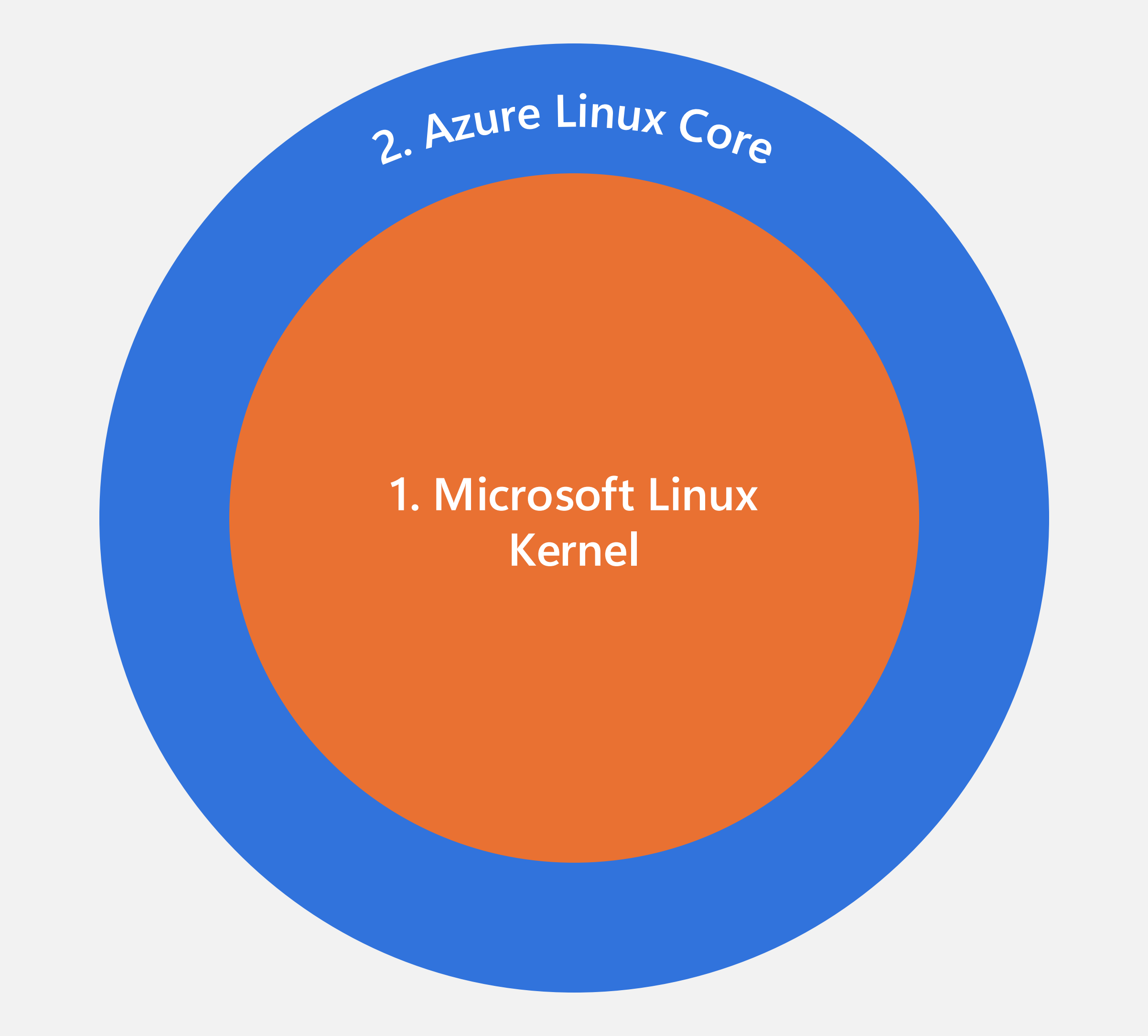
Microsoft releases Linux kernel to support WSL on Windows (announced at BUILD 2019)

2023

Microsoft releases Azure Linux for AKS Container Host

2009 | 2011 | 2012-2014 | 2015 | 2016-2018 | 2019-2023

What is Azure Linux?



1. Azure Linux Upstream LTS Linux Kernel

- Optimized for Hyper-V across cloud and edge, with Azure specific drivers for better performance.
- Hardened with secure defaults.
- Supports multiple silicon architectures used in Azure. E.g. x86_64, ARM64.

2. Azure Linux Core

- ~10,000 packages with fidelity (via RPM).
- Small footprint.

3. Around the clock protection from Microsoft

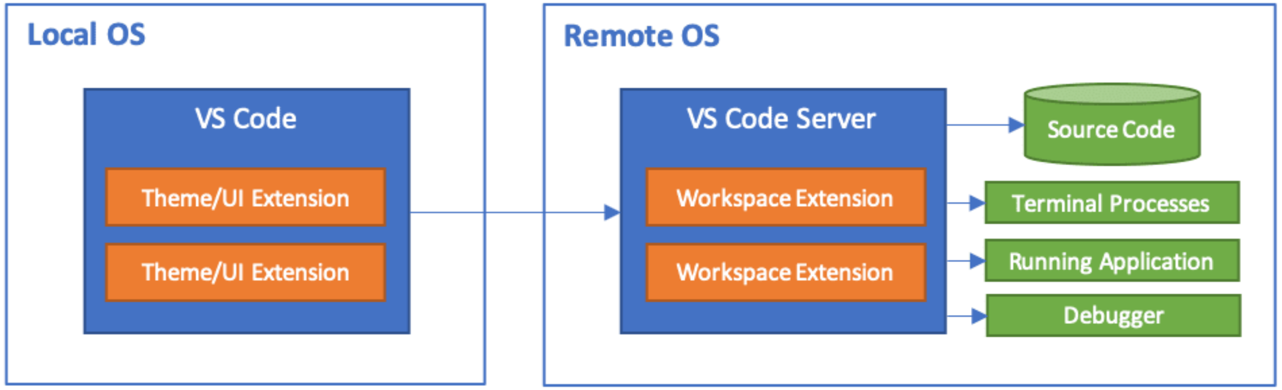
- Packages built from scratch to ensure supply chain security.
- Critical CVEs patches available within 5 days for commercially supported images.
- Image serviced monthly but on-demand updates for critical CVEs.

Installing and running .NET on Linux

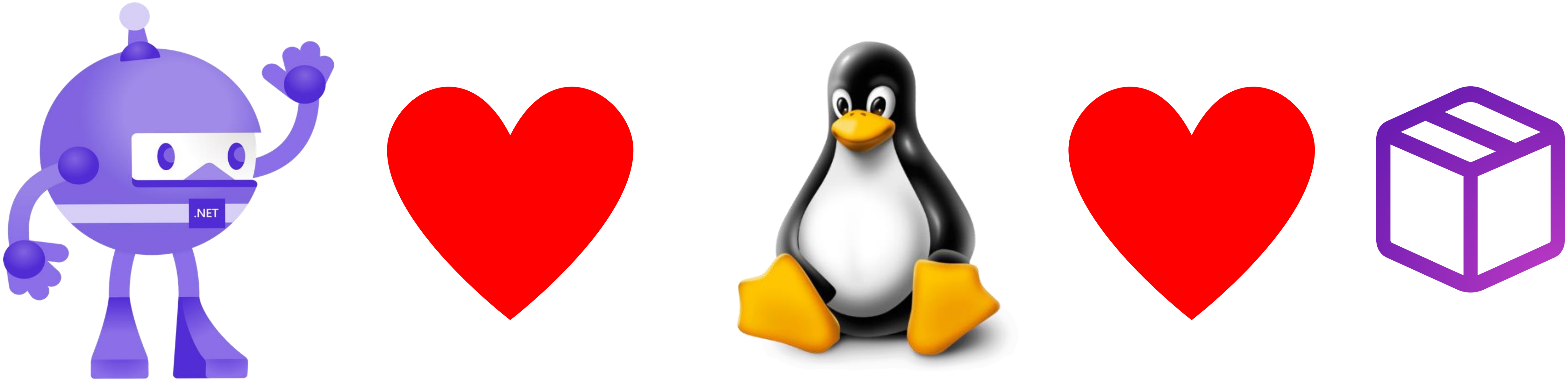
- Distros with official packages
  - Azure Linux 2.0 (CBL-Mariner)
  - Alpine
  - CentOS Stream
  - Fedora
  - Red Hat Enterprise Linux (RHEL)
  - Ubuntu
- Microsoft package repository
  - Azure Linux 3.0
  - Debian
  - openSUSE Leap
  - SUSE Enterprise Linux
  - …
- Snap
- Manual & scripted install
- Manual and scripted install supports
  - x86_64
  - aarch64
  - armv7
  - s390x
  - ppc64le
  - riscv64
- Microsoft and official package feed usually only supports the x64 architecture.
  - Scripted install is the easiest way to go if running Linux VMs on Apple Silicon

.NET on Linux Developer Experience

- From Linux Desktop
  - Install .NET SDKs and CLI tools
  - VSCode
- From MacOS
  - MacOS != Linux ☺
  - Parallels VMs (Apple Silicon, aarch64)
  - VSCode with Remote Development Extension
- From Windows
  - WSL
  - Linux VMs running anywhere
  - Full Visual Studio with Linux debugging
  - VSCode with Remote Development Extension



Demo


.NET

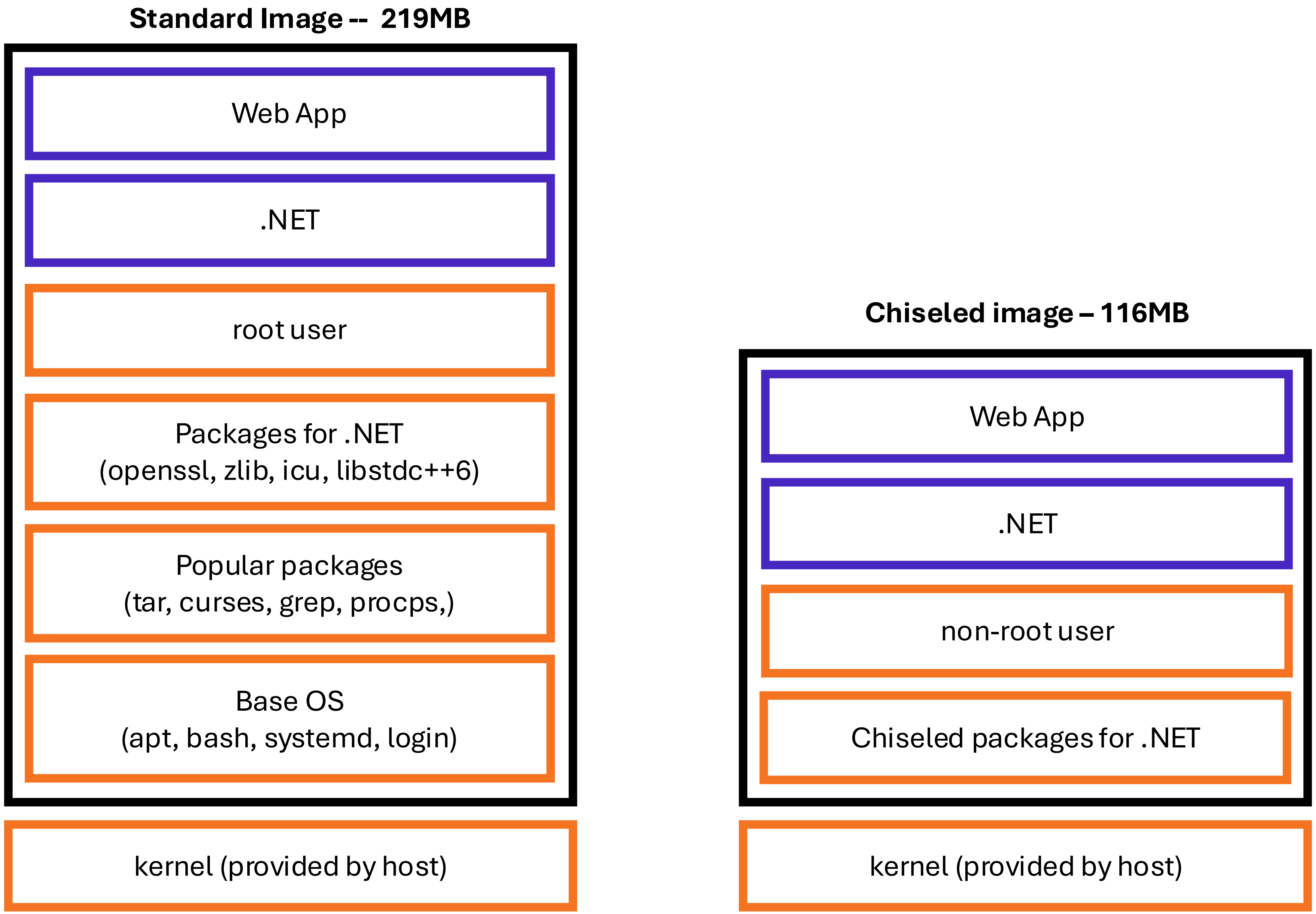
.NET & Containers

- Package an app with its dependencies and get that app to reliably run on any container host (Docker, K8s, Podman, etc.)
- Microsoft Artifacts Registry (MCR, mcr.microsoft.com) is a syndicate of Docker Hub
- .NET Linux images
  - SDKs and Runtimes (for building and running)
  - Alpine, Debian, Ubuntu, Azure Linux
- Containerize your app
  - Dockerfile
  - .NET SDK (dotnet publish /t:PublishContainer)

.NET differentiated images

- Target framework
- OS, version and proc architecture
- Image type
  - runtime, aspnet, sdk
- Image variant
  - *-distroless, *-chiseled
- Image features
  - *-aot, *-extra

Standard vs Chiseled .NET Images



Standard Image -- 219MB

Web App

.NET

root user

Packages for .NET
(openssl, zlib, icu, libstdc++6)

Popular packages
(tar, curses, grep, procps,)

Base OS
(apt, bash, systemd, login)

kernel (provided by host)

Chiseled image – 116MB

Web App

.NET

non-root user

Chiseled packages for .NET

kernel (provided by host)

What is distroless?

- The shell and package manager give the distro its feel and flexibility … to do *anything*.
- Distroless removes `bash` and `apt`, and anything else you don’t need. Do you really need `curses` and `tar`?
- Produces an appliance that does *one thing* not *anything*.

How do I update a distroless?

- Remember, this is all about containers.
- You don’t `ssh` into or `apt update/upgrade` them.
- You don’t `git pull` changes into them.
- You (regularly) re-build+deploy container images.
- You manage them with CI/CD like GitHub Actions.

Distroless: Hype or True Value?

The official openjdk image and distroless image have zero detected vulnerabilities, other images have multiple detected vulnerabilities.

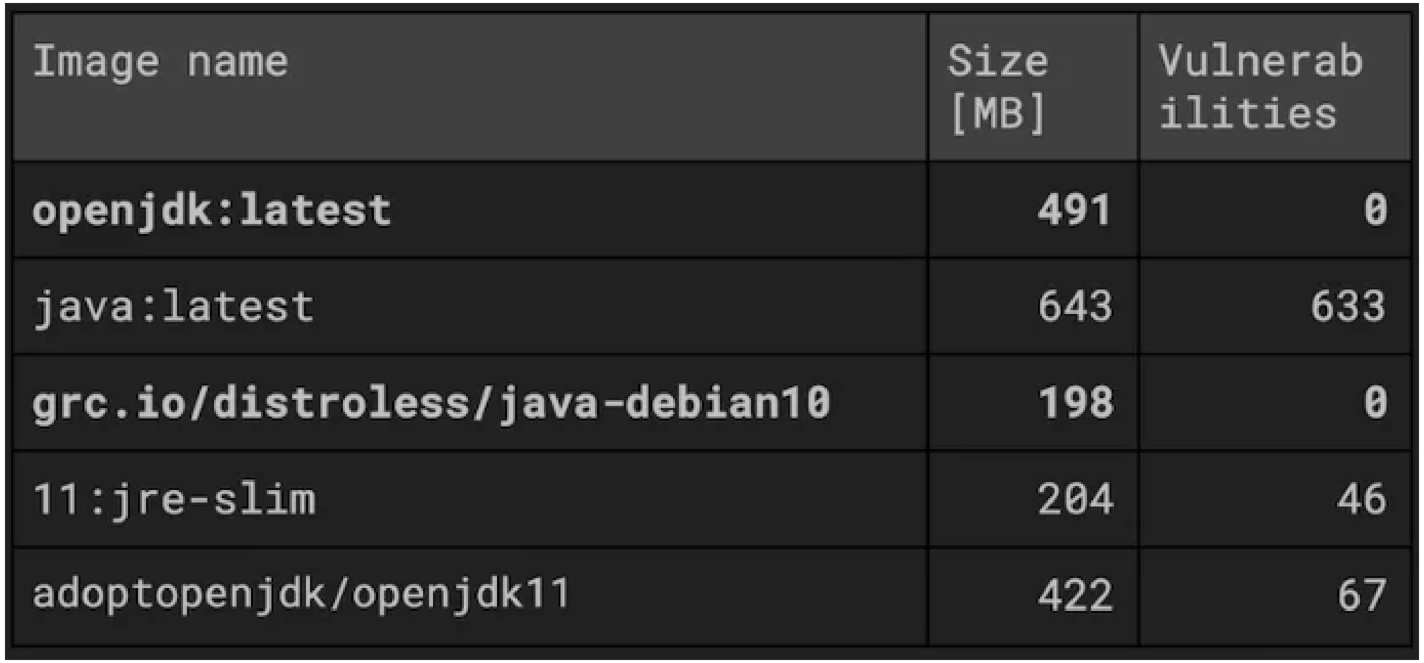


Image name	Size [MB]	Vulnerabilities
openjdk:latest	**491**	**0**
java:latest	643	633
grc.io/distroless/java-debian10	**198**	**0**
11:jre-slim	204	46
adoptopenjdk/openjdk11	422	67

Credit: https://hackernoon.com/distroless-containers-hype-or-true-value-2rfl3wat ; “grc” is a typo; should be “gcr”.

Azure services hosting containers

- Azure Kubernetes Service (AKS)

  Scale and orchestrate Windows & Linux containers using Kubernetes.
- Azure App Service

  Deploy web apps or APIs using containers in a PaaS environment.
- Azure Container Apps

  Run your container workloads without managing servers, orchestration, or infrastructure and leverage native support for Dapr and KEDA for observability and scaling to zero.
- Azure Container Instances

  Create individual containers in the cloud without any higher-level management services.
- Azure Batch

  Run repetitive compute jobs using containers.
- Azure Service Fabric

  Lift, shift, and modernize .NET applications to microservices using Windows & Linux containers.
- Azure Container Registry

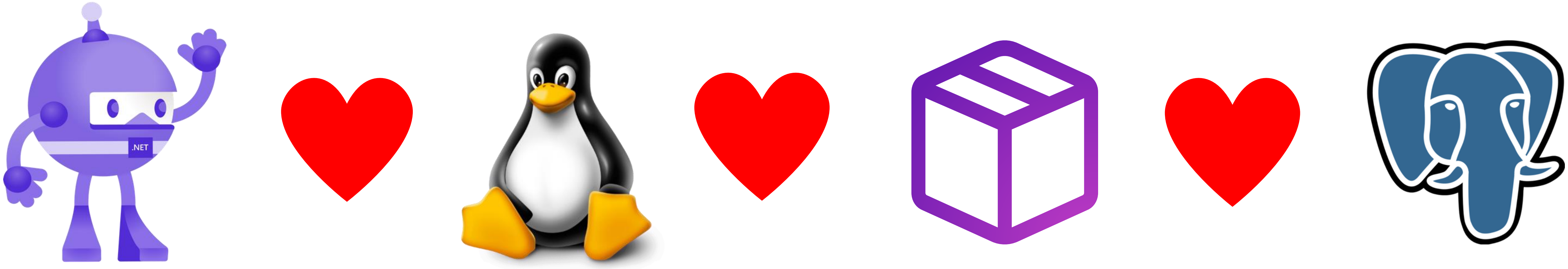
  Store and manage container images across all types of Azure deployments.

Demo

.NET & Containers

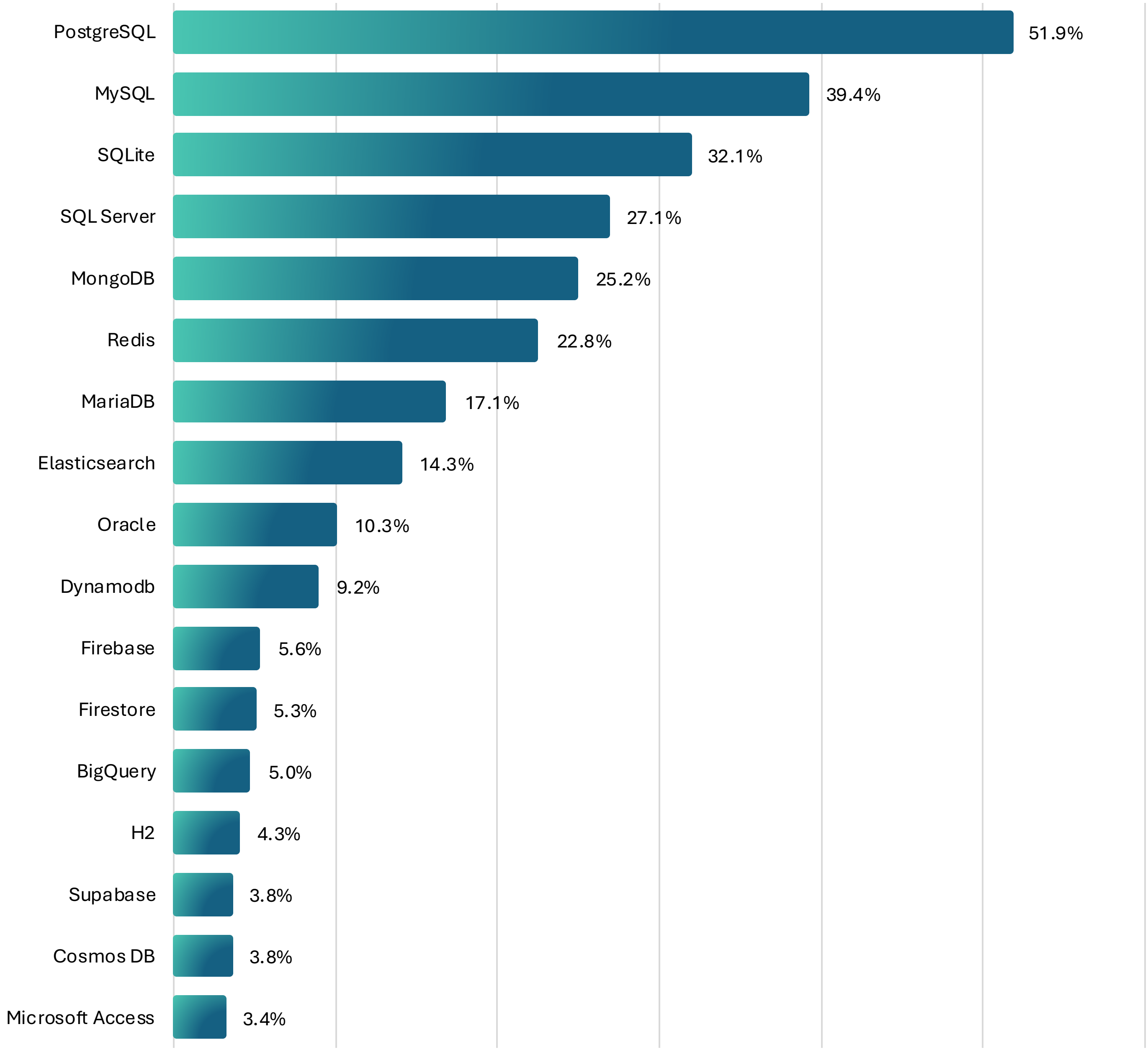
.NET Conference
Italia 2024

.NET


.NET

Postgres is the most popular database for professional developers

PostgreSQL extended lead 2024

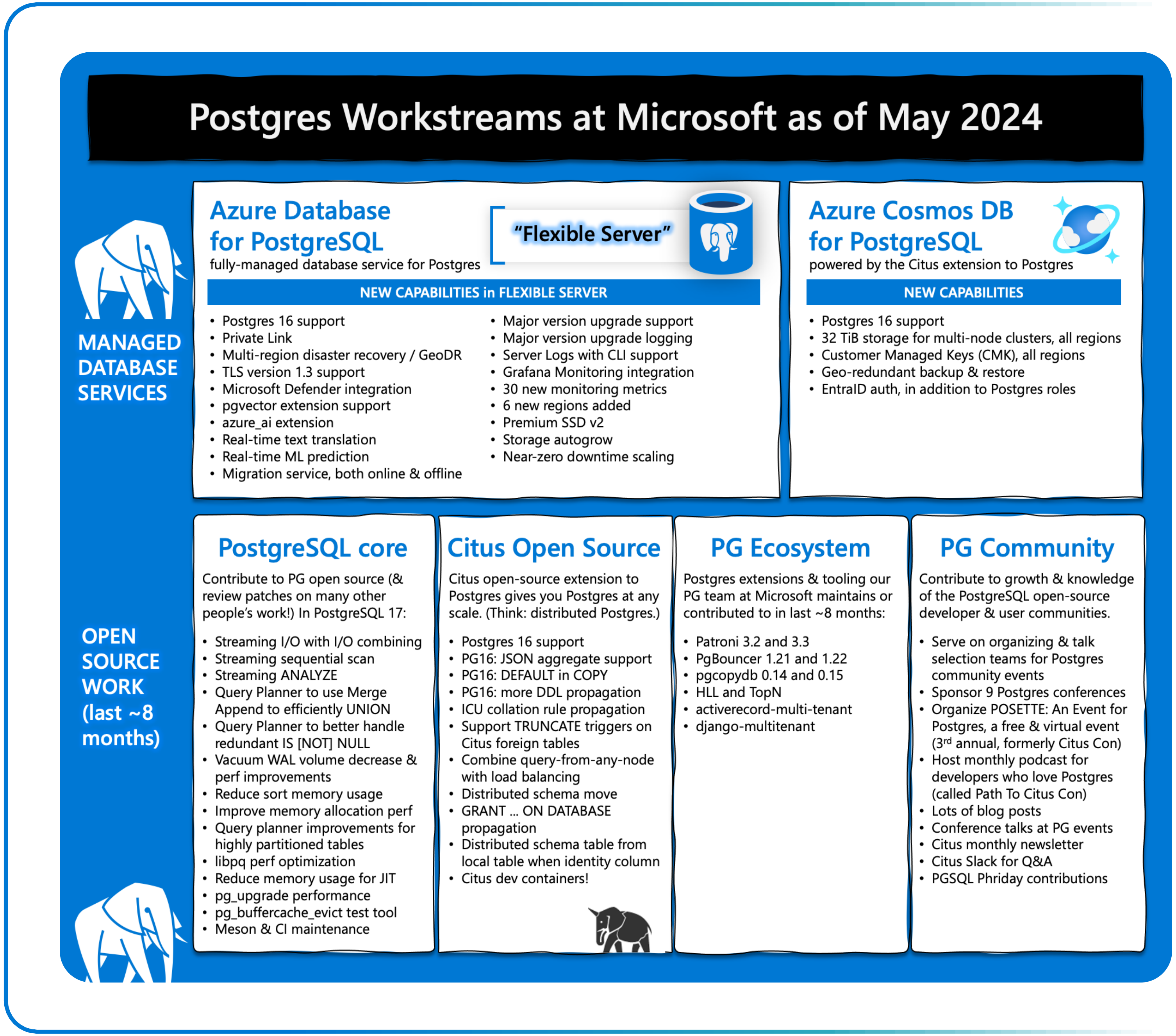


*Which **database environments** have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the database and want to continue to do so, please check both boxes in that row.)*

Source:
Stack Overflow Developer Survey 2024

Microsoft OSS investments extend beyond core PostgreSQL

https://aka.ms/blog-pg-at-microsoft



Postgres Workstreams at Microsoft as of May 2024

MANAGED DATABASE SERVICES

Azure Database for PostgreSQL

fully-managed database service for Postgres

"Flexible Server"

NEW CAPABILITIES in FLEXIBLE SERVER

- Postgres 16 support
- Private Link
- Multi-region disaster recovery / GeoDR
- TLS version 1.3 support
- Microsoft Defender integration
- pgvector extension support
- azure_ai extension
- Real-time text translation
- Real-time ML prediction
- Migration service, both online & offline
- Major version upgrade support
- Major version upgrade logging
- Server Logs with CLI support
- Grafana Monitoring integration
- 30 new monitoring metrics
- 6 new regions added
- Premium SSD v2
- Storage autogrow
- Near-zero downtime scaling

Azure Cosmos DB for PostgreSQL

powered by the Citus extension to Postgres

NEW CAPABILITIES

- Postgres 16 support
- 32 TiB storage for multi-node clusters, all regions
- Customer Managed Keys (CMK), all regions
- Geo-redundant backup & restore
- EntraID auth, in addition to Postgres roles

OPEN SOURCE WORK (last ~8 months)

PostgreSQL core

Contribute to PG open source (& review patches on many other people's work!) In PostgreSQL 17:

- Streaming I/O with I/O combining
- Streaming sequential scan
- Streaming ANALYZE
- Query Planner to use Merge Append to efficiently UNION
- Query Planner to better handle redundant IS [NOT] NULL
- Vacuum WAL volume decrease & perf improvements
- Reduce sort memory usage
- Improve memory allocation perf
- Query planner improvements for highly partitioned tables
- libpq perf optimization
- Reduce memory usage for JIT
- pg_upgrade performance
- pg_buffercache_evict test tool
- Meson & CI maintenance

Citus Open Source

Citus open-source extension to Postgres gives you Postgres at any scale. (Think: distributed Postgres.)

- Postgres 16 support
- PG16: JSON aggregate support
- PG16: DEFAULT in COPY
- PG16: more DDL propagation
- ICU collation rule propagation
- Support TRUNCATE triggers on Citus foreign tables
- Combine query-from-any-node with load balancing
- Distributed schema move
- GRANT … ON DATABASE propagation
- Distributed schema table from local table when identity column
- Citus dev containers!

PG Ecosystem

Postgres extensions & tooling our PG team at Microsoft maintains or contributed to in last ~8 months:

- Patroni 3.2 and 3.3
- PgBouncer 1.21 and 1.22
- pgcopydb 0.14 and 0.15
- HLL and TopN
- activerecord-multi-tenant
- django-multitenant

PG Community

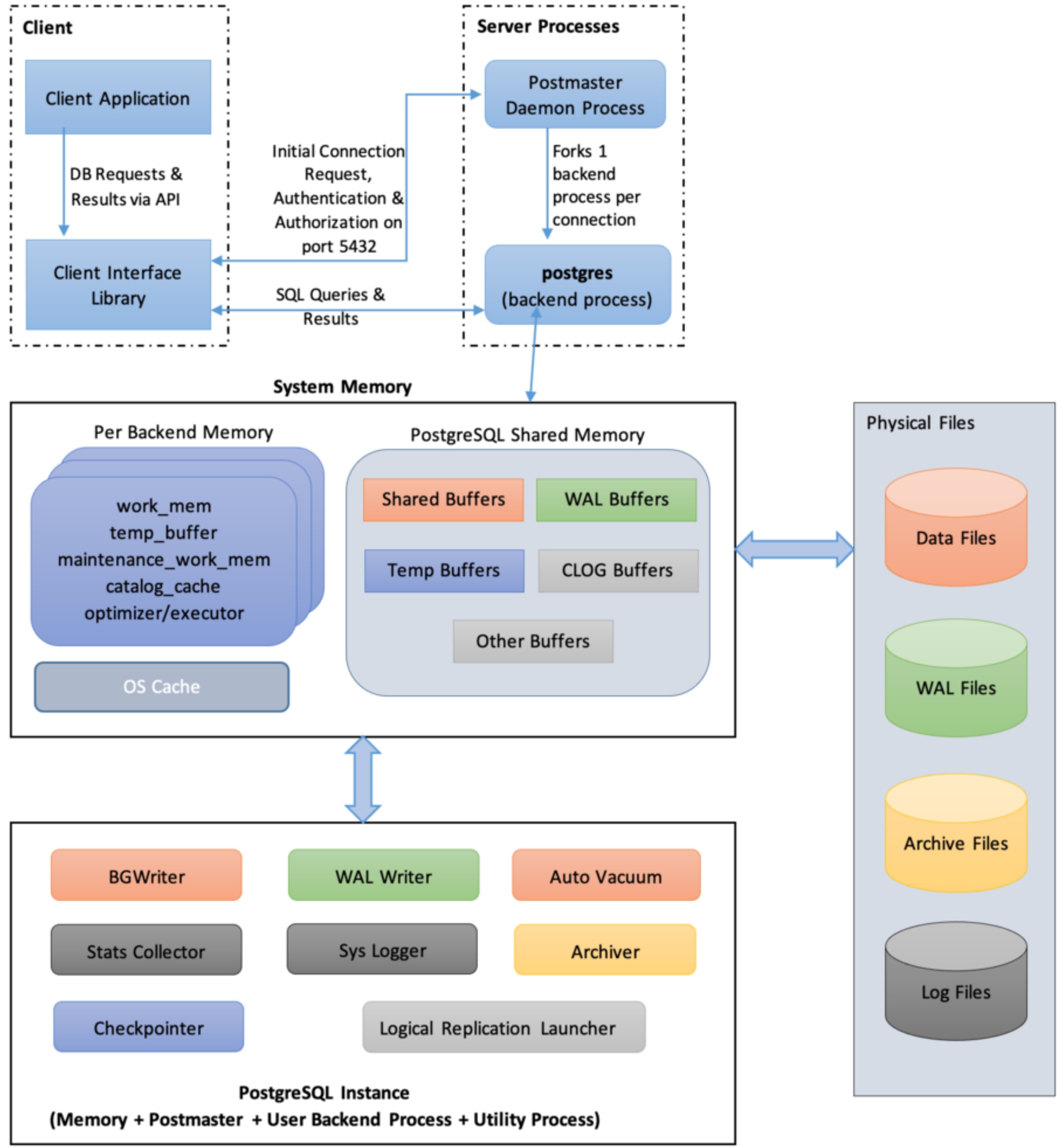
Contribute to growth & knowledge of the PostgreSQL open-source developer & user communities.

- Serve on organizing & talk selection teams for Postgres community events
- Sponsor 9 Postgres conferences
- Organize POSETTE: An Event for Postgres, a free & virtual event (3rd annual, formerly Citus Con)
- Host monthly podcast for developers who love Postgres (called Path To Citus Con)
- Lots of blog posts
- Conference talks at PG events
- Citus monthly newsletter
- Citus Slack for Q&A
- PGSQL Phriday contributions

Why PostgreSQL?

- Community driven OSS , portable
  - Great Community Support and resources
  - Standard compliance
- Designed on Linux/Unix
  - “works” on Windows
  - Multi-process, shared memory
- Runs on all major proc architectures
- Extensibility by design from the core
  - Well-defined APIs to plug-in new implementations (types, operators, etc.)

PostgreSQL architecture



Client

Client Application

DB Requests & Results via API

Client Interface Library

Initial Connection Request, Authentication & Authorization on port 5432

SQL Queries & Results

Server Processes

Postmaster Daemon Process

Forks 1 backend process per connection

postgres (backend process)

System Memory

Per Backend Memory

work_mem
temp_buffer
maintenance_work_mem
catalog_cache
optimizer/executor

OS Cache

PostgreSQL Shared Memory

Shared Buffers

WAL Buffers

Temp Buffers

CLOG Buffers

Other Buffers

Physical Files

Data Files

WAL Files

Archive Files

Log Files

BGWriter

WAL Writer

Auto Vacuum

Stats Collector

Sys Logger

Archiver

Checkpointer

Logical Replication Launcher

PostgreSQL Instance
(Memory + Postmaster + User Backend Process + Utility Process)

Why PG architecture is relevant for app dev?

- Multi-process architecture makes connection management more expensive than other databases (e.g. max connection default = 100)
  - App-level connection pooling is critical
- MVCC and optimistic concurrency by default
  - Manage concurrency tokens instead of locking
  - Framework like EF Core can use system columns like xmin to track latest updating transaction for a tuple

PostgreSQL Fundamentals

Deployment

- Supported natively on any operating system and proc architecture
  - Local Service or Daemon
  - Containers
  - Kubernetes
- Main distros' package managers
- PostgreSQL.org repositories
- Source code (GitHub or https://git.postgresql.org)
- Config files
  - General service
  - Authentication
  - Identities

Key characteristics

- Multi Version Concurrency Control (MVCC)
- Advanced data types (arrays, JSONB, geo, vectors etc.)
- Advanced indexing (GiST, GIN, BRIN, Bloom, etc.)
- Advanced programmability (PL/pgSQL, Perl, Python, Java, etc.)
- High availability with streaming replication
- Logical replication for data integration scenarios

Tooling

- Standard Unix/Linux tools
- Cumulative Statistics System
  - Collection configuration parameters
  - Control functions
  - Predefined views
- Lock contention: pg_locks
- Progress reporting
- pg_stat_statement extension
- Troubleshooting
  - Log analyzer: pgBadger

- Client tools
  - psql
  - pgAdmin
  - Azure Data Studio
  - DBeaver
- PostgreSQL client and server apps
  - reindexdb, vacuumdb, pg_dump, pg_waldump, etc
- Benchmarking
  - pgbench
  - HammerDB
- 3rd party
  - Dev & test
  - Monitoring
  - Management

Programmability

- Procedures, functions and triggers
- Schema and Programmability
    - Table inheritance
    - Declarative partitioning
    - Pattern matching ILIKE/SIMILAR TO Regex/ POSIX Regex.
- JSON
    - Rich Set of JSON operators
    - Expansive JSON functions
    - Indexing to support pattern matching

- Spatial data types
    - Geometry, geography, raster
    - Indexes
        - Rtree, quadtree
    - Functions
        - ST_Distance, ST_Area, ST_GeometryType,ST_Intersection….
    - Related extensions
- Foreign Data Wrappers
    - SQL Server
    - Oracle
    - …

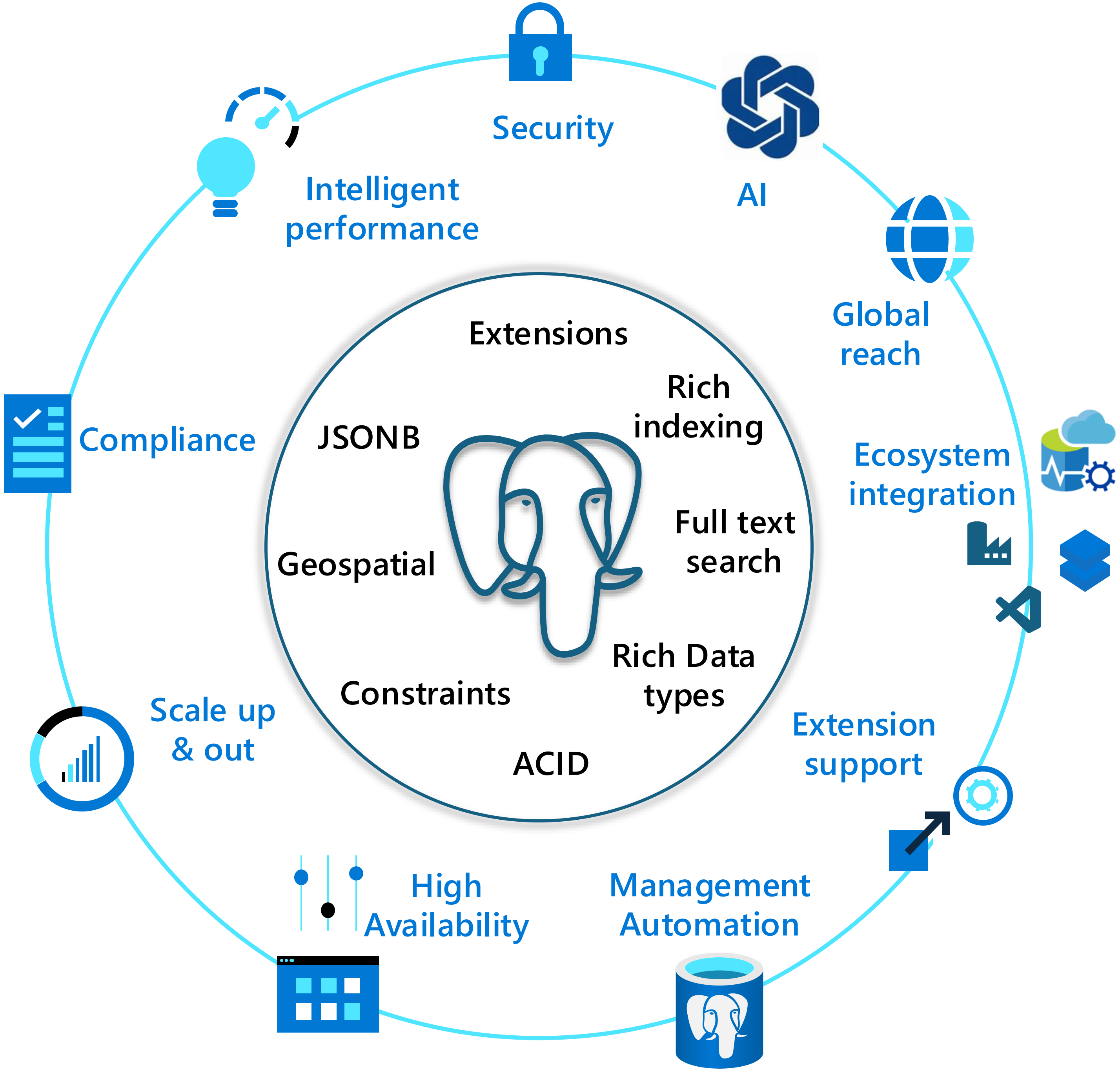
Npgsql - .NET Access to PostgreSQL

- ADO.NET Data Provider for PostgreSQL
    - 100% C# code
    - Free and open source (key maintainers in .NET Data team)
    - PostgreSQL wire protocol for apps written in C#, Visual Basic, F#
- Entity Framework Core provider
    - Full support for complex data types (e.g. JSON, Vector, Spatial, Arrays, Ranges, etc.)
    - PostgreSQL-specific conventions and behaviors
    - PostgreSQL-specific indexes and operator classes
    - Concurrency tokens

Npgsql - Data design and lifecycle in your app

- Database creation with EnsureCreated()
  - Administrative db (*options.UseAdminDatabase("my_admin_db")*)
  - Database template (*modelBuilder.UseDatabaseTemplate("my_template_db")*)
- Migrations
  - *dotnet ef tool*
  - *dbContext.Database.Migrate();*
- Seeding
  - UseSeeding and UseAsyncSeeding
  - Called as part of *EnsureCreated()*, *Migrate()* and *dotnet ef database update* commands
- Database first
  - EF Tools Scaffolding
- Support for Logical Replication decoding
  - Reactive scenarios (CDC, outbox to a queue, microservices notifications, etc.)

Azure builds upon PostgreSQL



Demo

PostgreSQL

.NET Conference Italia 2024

.NET

.NET – Linux – Containers – PostgreSQL

Better together!

- From your dev box => to on-prem DC => to all cloud providers
- Simplified inner loop
  - Develop and test locally -> push to Container Registry and deploy everywhere
- Structured outer loop
  - Build and release management pipelines on Azure DevOps or GitHub Actions
- .NET Aspire
  - Build, test, and deploy apps seamlessly from code to cloud

.NET Aspire

Build, test, and deploy apps seamlessly from code to cloud

- Streamlined Inner-Loop
- Developer Dashboard
- Integrations
- Deployment

Extensible, OpenTelemetry Built-in, & Ready for Any Cloud

.NET Aspire

Get started

Open-source

Templates

Integrations

To build

Service discovery

Developer dashboard

Logs, metrics,
distributed traces

To deploy

Single command run

App topology in C#

Cloud deployment

.NET Aspire 9.0

Azure Functions preview support

More secure defaults

"WaitFor" resources to spin up

Start/Stop Resources

Improved Azure Configuration

Simplified acquisition

Improved Azure Container Apps integration

Persistent Containers

Support for .NET 8 & .NET 9

OpenAI

OpenAI Integration

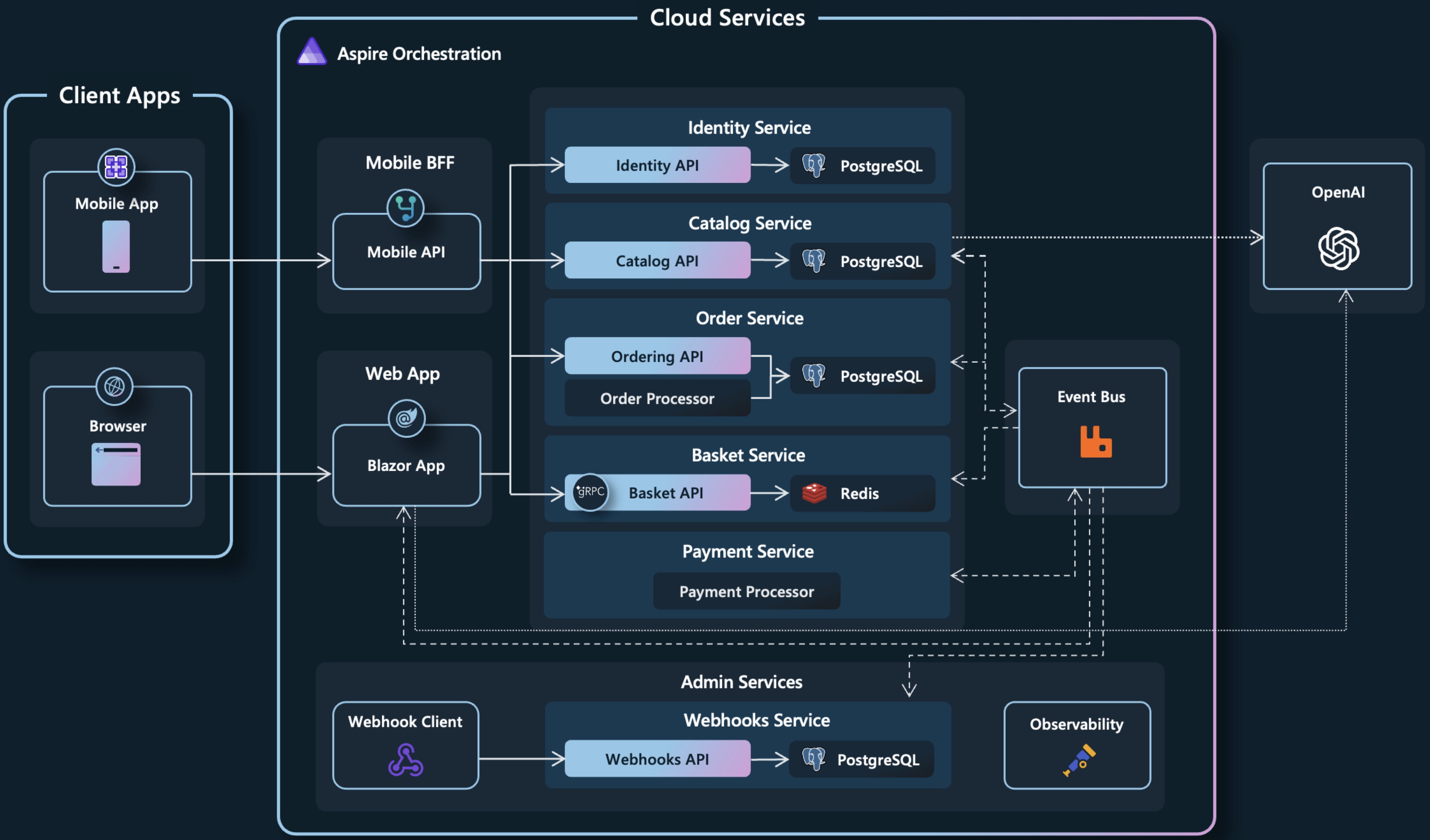
AWS Stable Integrations

Visual Studio & C# Dev Kit enhancements

.NET Aspire Community Toolkit

https://github.com/dotnet/eShop

eShop reference application



Demo

Better together!

.NET Conference Italia 2024

.NET

Silvano Coriani
PostgreSQL @ Azure
linkedin.com/in/scoriani/
github.com/scoriani

aspitalia.com

UGIdotNET

Slide e materiale su
https://www.dotnetconference.it/

.NET Conference
Italia 2024

.NET