



openGauss+Wasm 构建安全高效的UDF执行引擎

贺国锋
openGauss社区运营经理

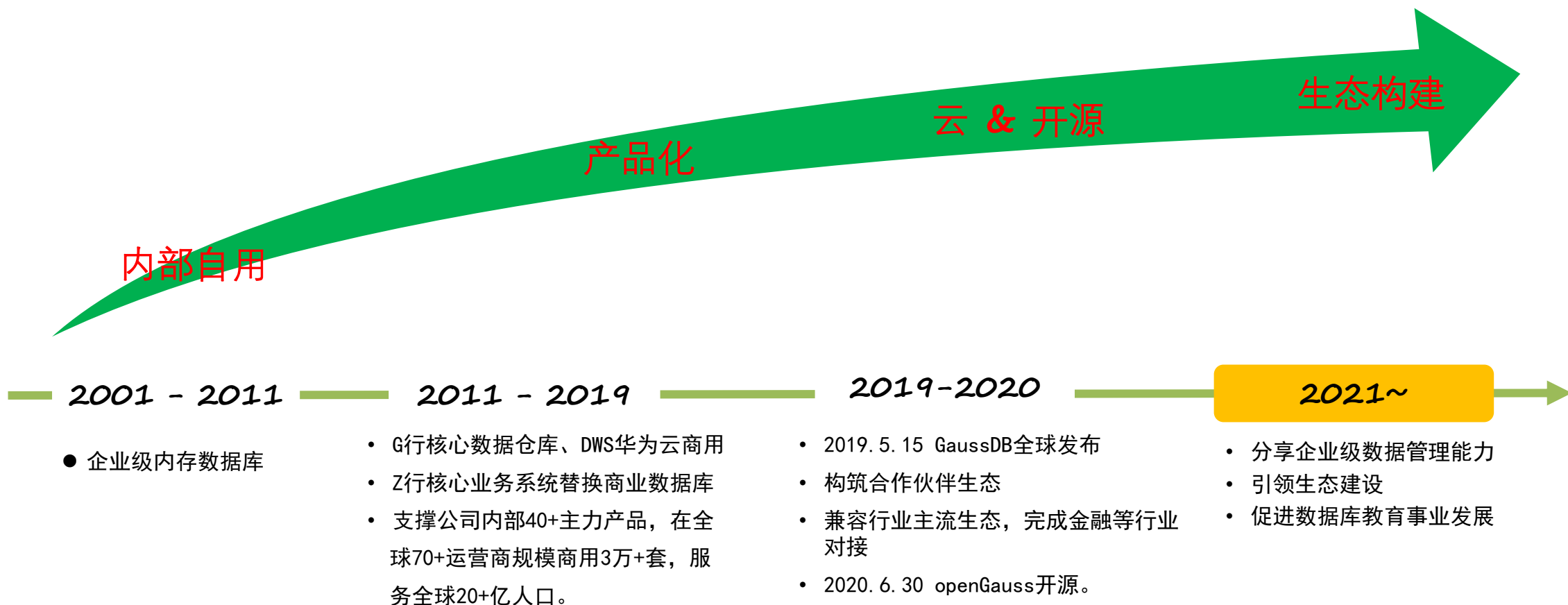
目录

- openGauss介绍
- Wasm介绍
- openGauss + Wasm

openGauss介绍

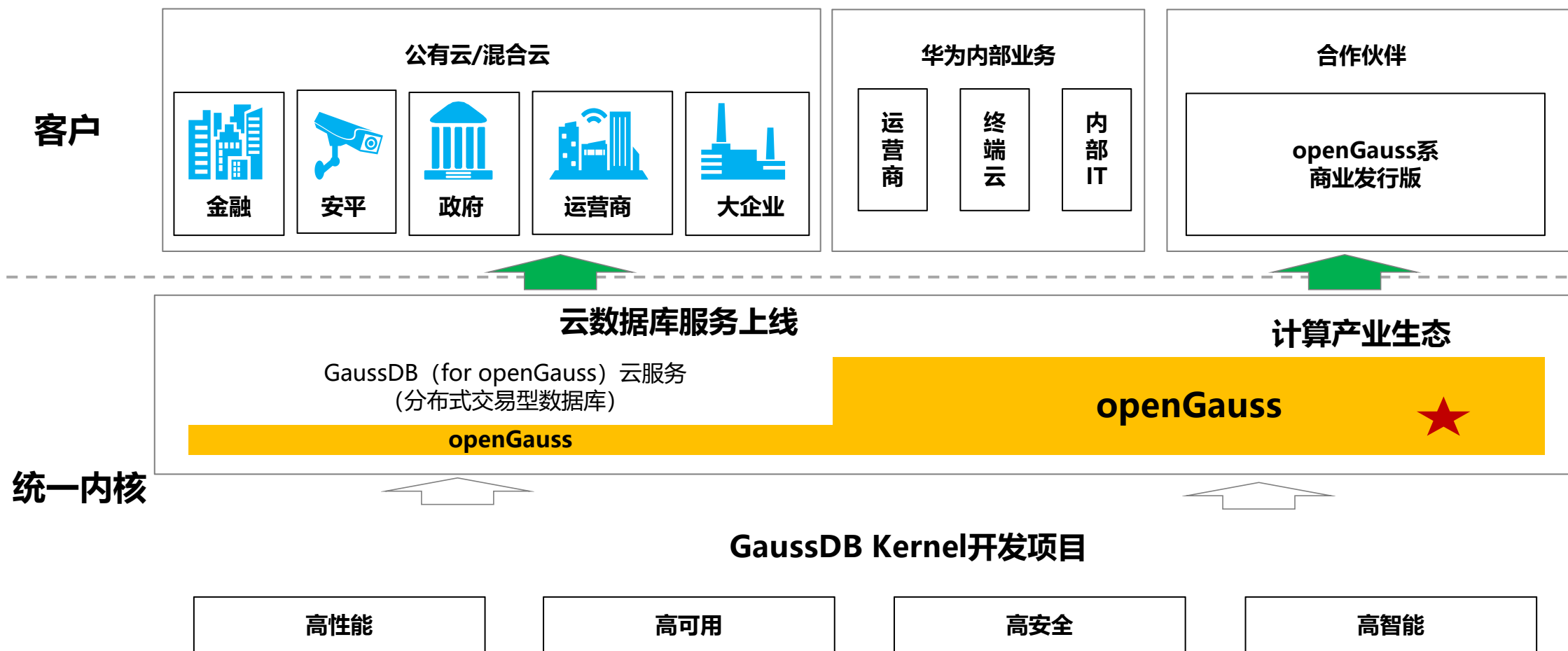
华为GaussDB演进历程

内部自用孵化阶段 → 联创产品化阶段 → 共建生态



openGauss产品：商用+自用+开源相结合，内核将长期演进

华为公司内部配套、公有云的GaussDB、开源openGauss 共代码基线





openGauss 定位

把企业级数据库能力带给用户和伙伴

价值

openGauss提供面向多核的极致性能、全链路的业务和数据安全、基于AI的调优和高效运维的能力，全面友好开放，携手伙伴共同打造全球领先的企业级开源关系型数据库；

关键特性

高性能

- 两路鲲鹏性能**150万tpmC**；
- 面向多核架构的并发控制技术；
- NUMA-Aware数据结构；
- SQL-Bypass智能选路执行技术；
- 面向实时高性能场景的内存引擎；
- Inplace-Update引擎

高可用 & 高安全

- 业务无忧，故障切换时间RTO<10s；
- 基于Paxos协议的高可用
- 精细安全管理：细粒度访问控制、多维度审计；
- 全方位数据保护：存储&传输&导出加密、动态脱敏、全密态计算、防篡改；

易运维

- AI自治运维：参数推荐、慢SQL诊断、索引推荐、趋势预测及异常检测等；
- WDR报告：多维性能自监控视图，实时掌控系统性能表现；
- 库内AI引擎：通过简易SQL接口提供库内机器学习算法的训练和推理；

全开放

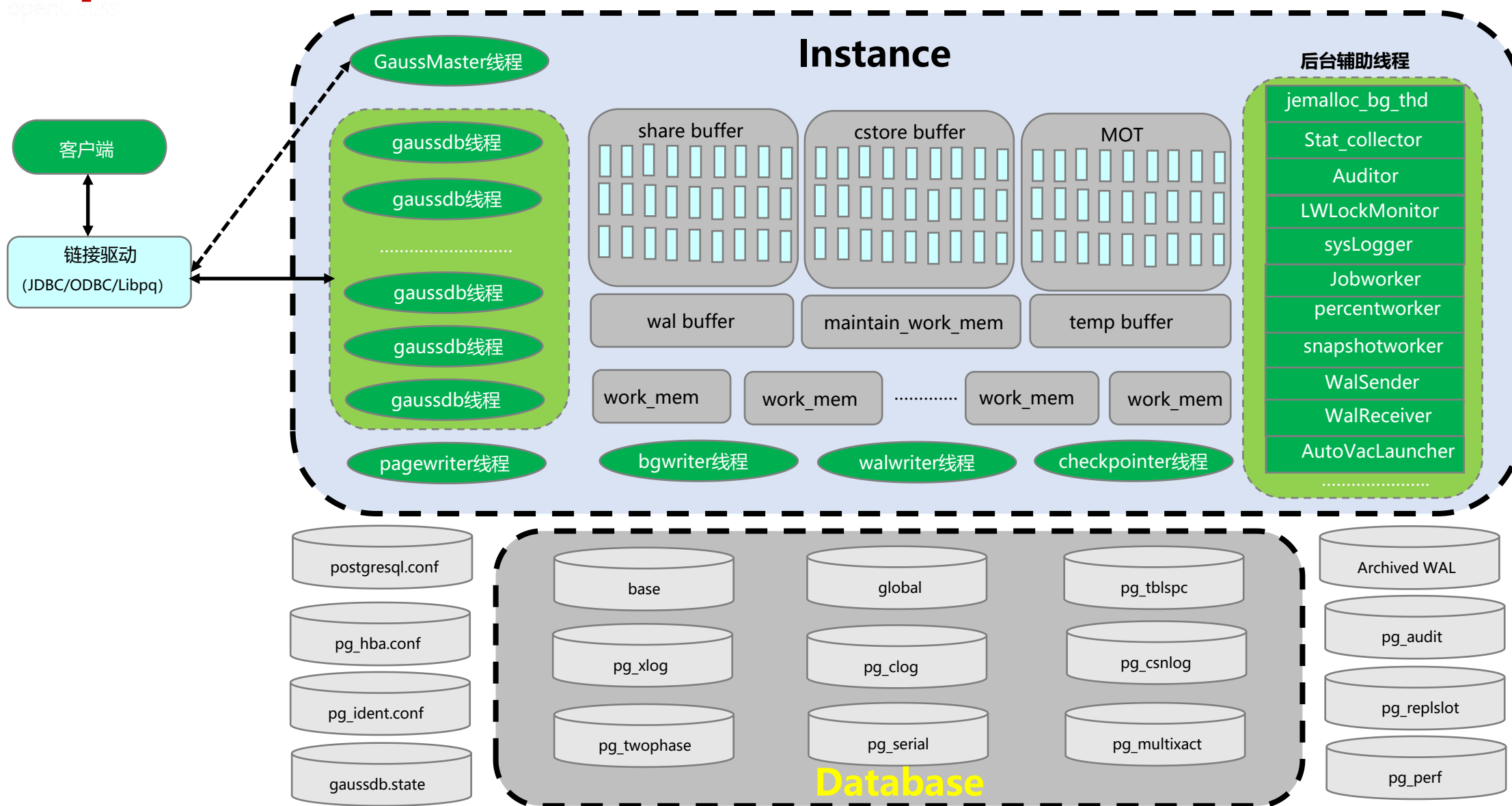
- 采用木兰宽松许可证协议，允许对代码自由修改、使用、引用；
- 数据库内核能力完全开放；
- 开放运维监控、开发和迁移工具；
- 开放伙伴认证、培训体系及高校课程



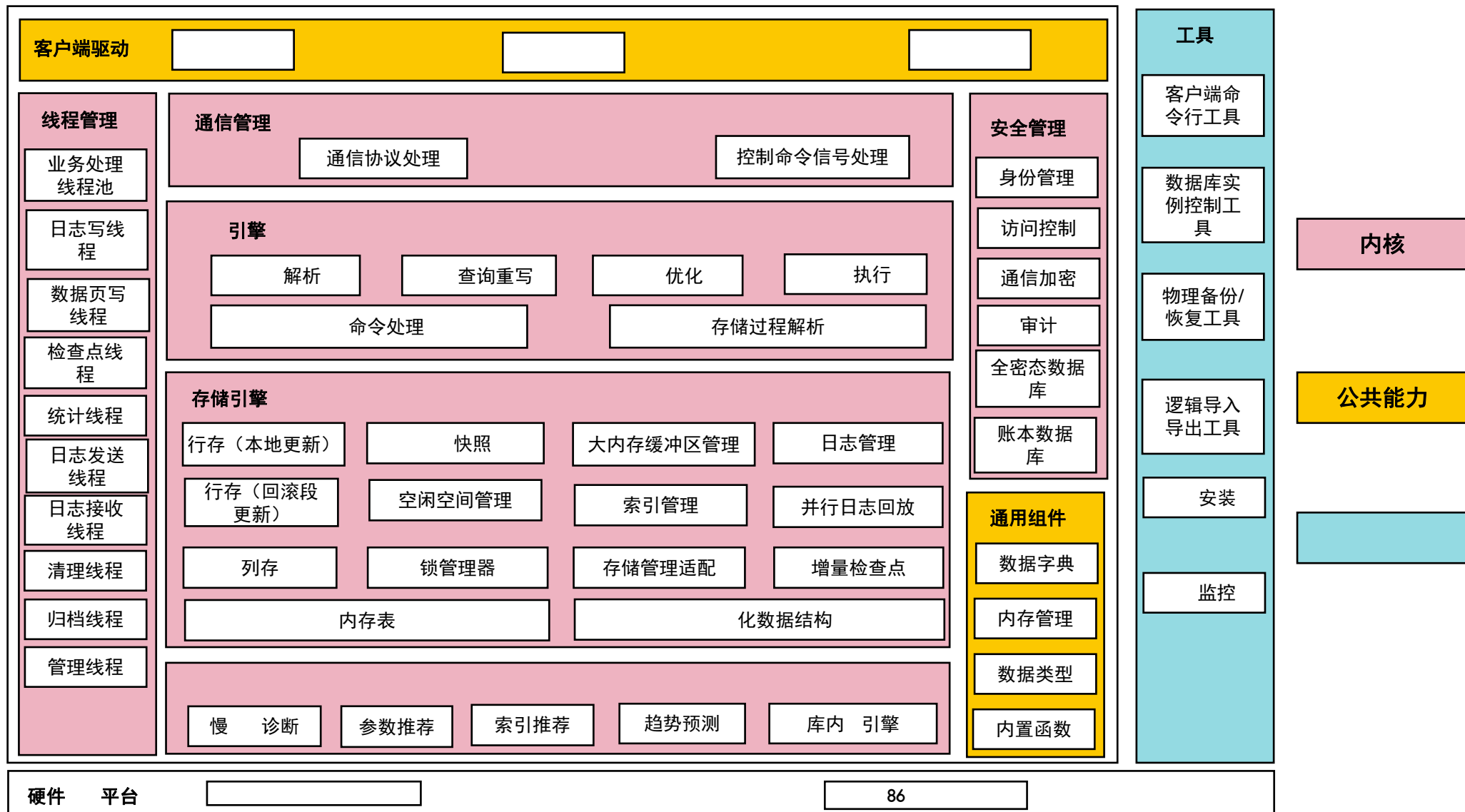
openGauss

<https://opengauss.org>

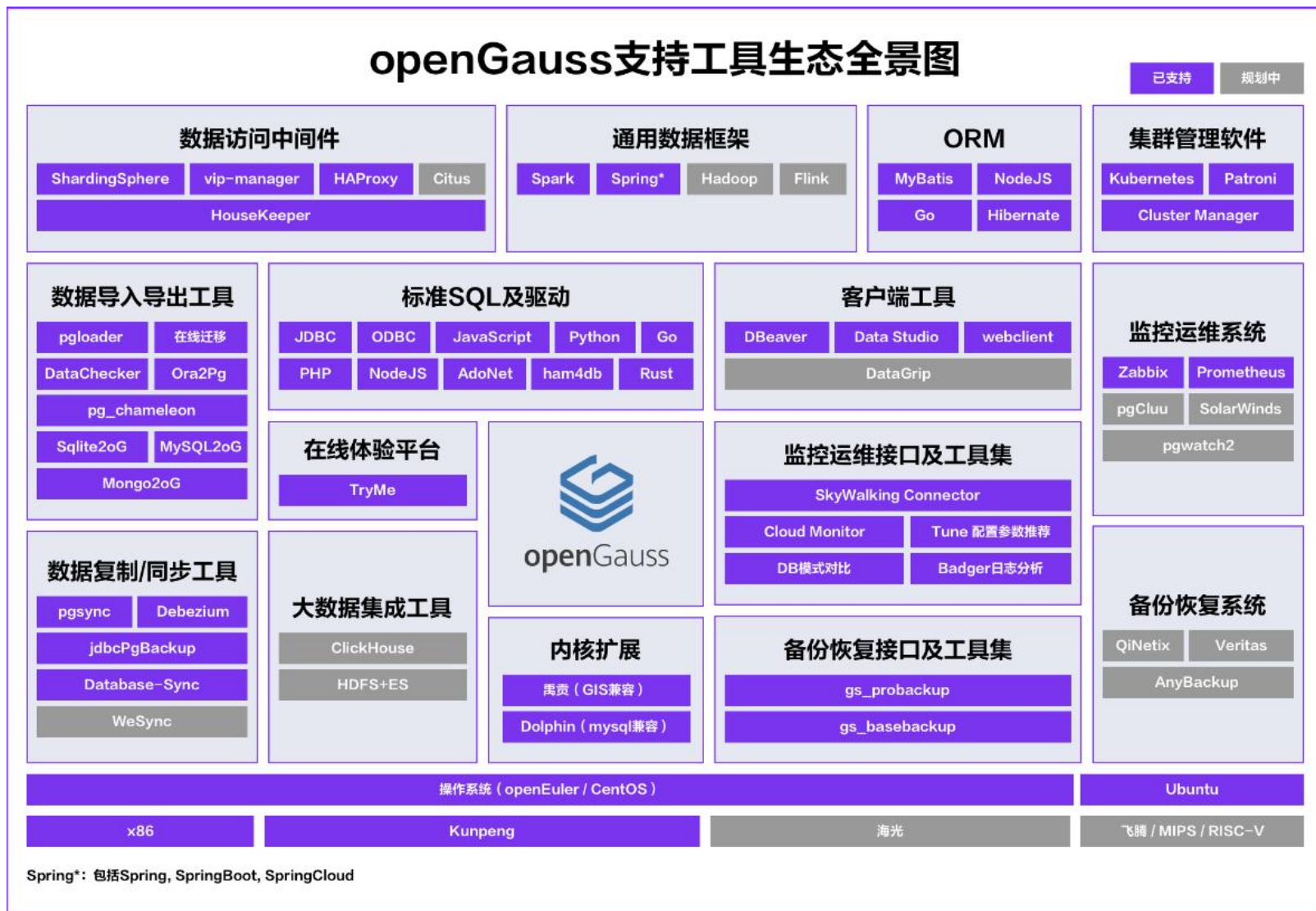
openGauss 体系结构



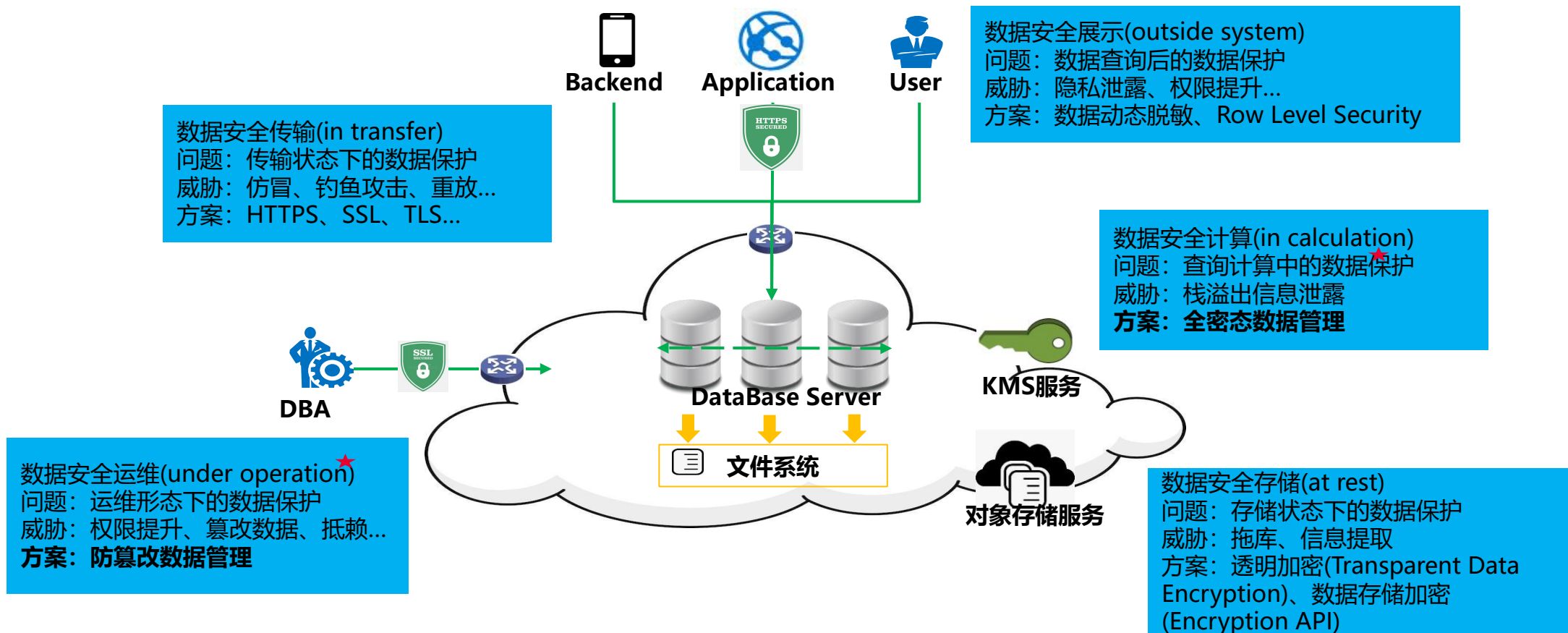
openGauss 逻辑模块



openGauss 工具生态全景图



openGauss未来技术方向：更智能、更安全、更高效



Wasm介绍

Wasm定义及其发展

WebAssembly (abbreviated Wasm) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

Wasm是一种基于栈式虚拟机的二进制指令格式，旨在为众多编程语言提供一种可移植的编译目标，方便应用在web客户端及服务器上的部署运行

2015

- 4月，WebAssembly Community Group成立
- 6月WebAssembly第一次以WCG官方名字对外公布

2017

- 2月，LOGO确定
- 2月，Firefox、Chrome、Edge、WebKit在WebAssembly的MVP（最小可用版本）标准实现上达成共识
- 8月，W3C WebAssembly Working Group成立，正式成为W3C技术标准

2019

- 11月，Mozilla、Fastly、Intel与Red Hat宣布成立字节码联盟Bytecode Alliance，旨在通过协作实施Wasm标准和Wasi标准，以完善WebAssembly在浏览器之外的生态
- 12月，WebAssembly成为W3C推荐标准，与HTML、CSS和JavaScript一起成为web的四种语言

2021

- 1月，Wasmer1.0版本发布
- 8月，WasmEdge 0.8.0版本发布

2022

- 9月，可用于生产环境的Wasmtime1.0版本发布
- 10月，Docker发布Wasm技术预览版本Docker+Wasm

Efficient and fast

The Wasm [stack machine](#) is designed to be encoded in a size- and load-time-efficient [binary format](#). WebAssembly aims to execute at native speed by taking advantage of [common hardware capabilities](#) available on a wide range of platforms.

Open and debuggable

WebAssembly is designed to be pretty-printed in a [textual format](#) for debugging, testing, experimenting, optimizing, learning, teaching, and writing programs by hand. The textual format will be used when [viewing the source](#) of Wasm modules on the web.

Safe

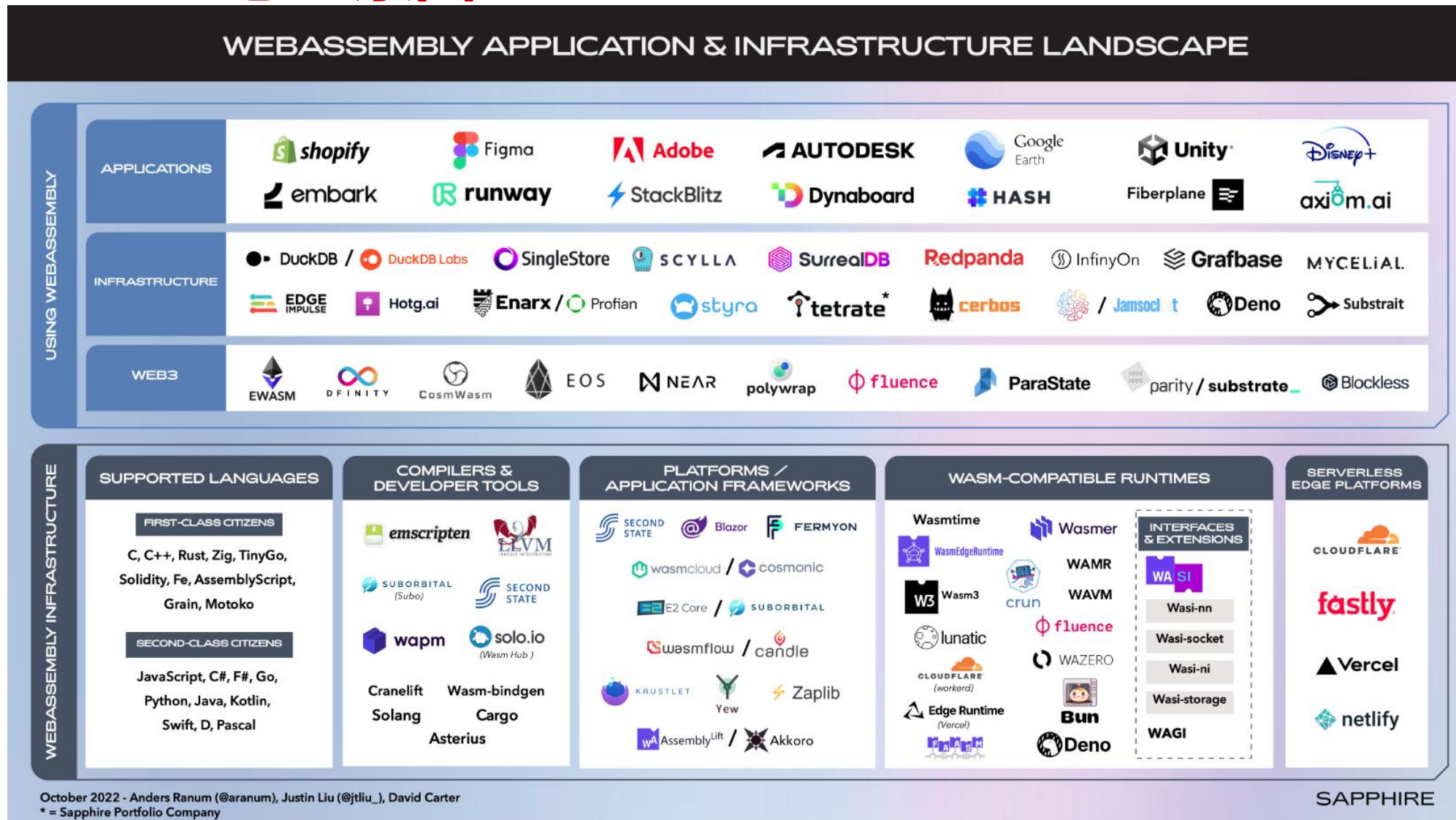
WebAssembly describes a memory-safe, sandboxed [execution environment](#) that may even be implemented inside existing JavaScript virtual machines. When [embedded in the web](#), WebAssembly will enforce the same-origin and permissions security policies of the browser.

Part of the open web platform

WebAssembly is designed to maintain the versionless, feature-tested, and backwards-compatible [nature of the web](#). WebAssembly modules will be able to call into and out of the JavaScript context and access browser functionality through the same Web APIs accessible from JavaScript. WebAssembly also supports [non-web](#) embeddings.

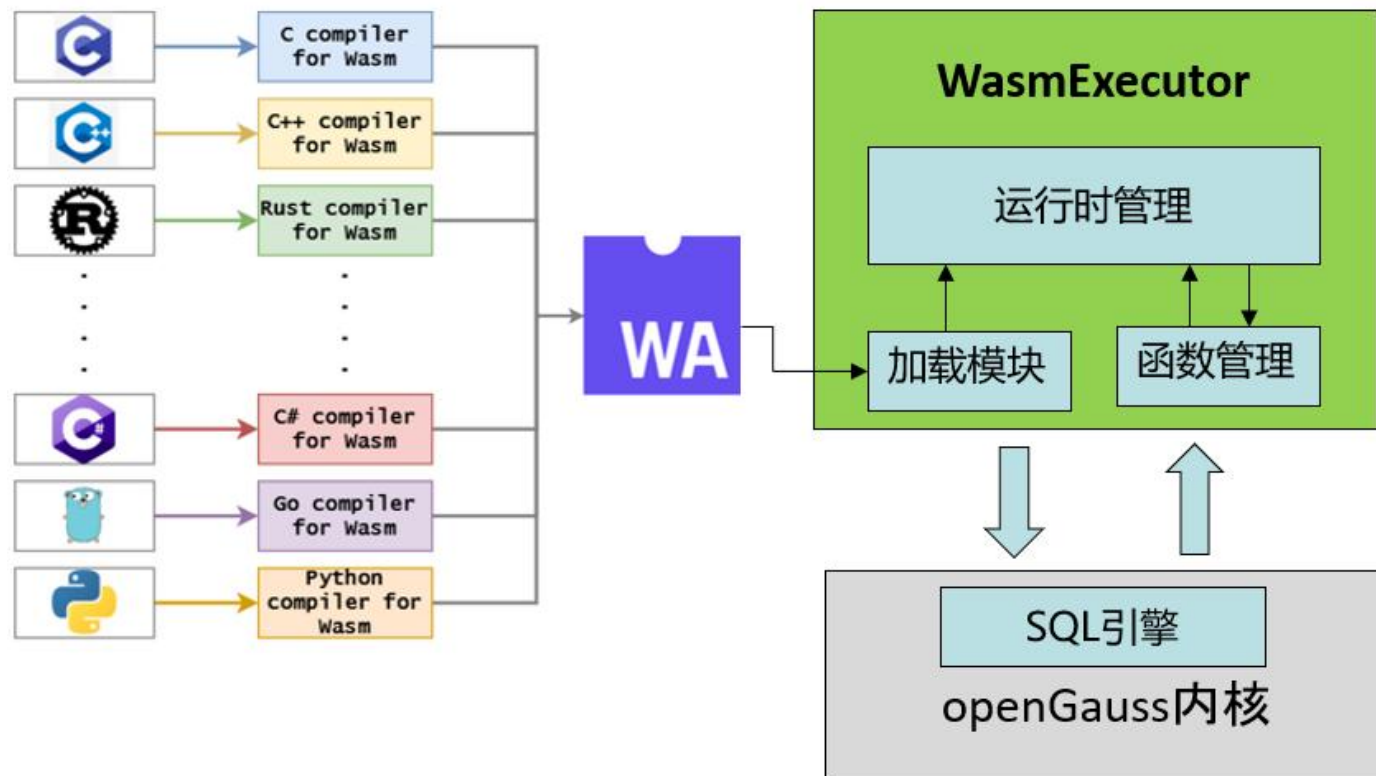
- 高效
- 安全
- 开放
- 标准

Wasm生态全景图



openGauss + Wasm

openGauss 体系结构



- 可插拔：基于openGauss插拔式接口，实现引擎的动态加载和卸载，避免侵入内核，独立升级和演进。
- 支持多语言：兼容Illum生态，支持使用Go/Rust/Python等众多语言开发
- 跨平台：开发的UDF函数可同时运行在x86和arm上
- 简便易用，原生SQL体验：使用引擎提供的SQL函数即可批量导入Wasm函数，后续直接使用SQL调用

加载函数

wasm_new_instance
wasm_new_instance_wat

元信息表

wasm.instances
wasm.exported_functions

WasmExecutor: 加载执行流程



WasmExecutor: 调用执行流程





openGauss容器镜像

opengaussofficial / opengauss Contains: Image Last pushed: 6 days ago	Not Scanned	☆ 0	↓ 7	Public
opengaussofficial / opengauss-wasmtime Contains: Image Last pushed: 7 days ago	Not Scanned	☆ 0	↓ 12	Public
opengaussofficial / opengauss-dev Contains: Image Last pushed: 9 days ago	Not Scanned	☆ 1	↓ 37	Public
opengaussofficial / opengauss-webclient Contains: Image Last pushed: 15 days ago	Not Scanned	☆ 0	↓ 5	Public

镜像仓库:

<https://hub.docker.com/u/opengaussofficial>

```
docker pull opengaussofficial/opengauss-wasmtime:0.1.0
```

```
docker run -it opengaussofficial/opengauss-wasmtime:0.1.0
```



操作示例

```
[opengauss@489620fe2d9c ~]$ gsql -d postgres
gsql ((openGauss 3.1.0 build 2fa6cc8d) compiled at 2022-11-30 18:42:47 commit 0 last mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.
```

```
openGauss=# create extension wasm_executor;
CREATE EXTENSION
openGauss=#
```

```
[opengauss@489620fe2d9c ~]$ ls -lrt
total 40
drwxr-xr-x 4 opengauss opengauss 4096 Sep 27 04:36 wasmtime
-rw-r--r-- 1 opengauss opengauss 569 Nov 9 19:45 bashrc
-rw-r--r-- 1 opengauss opengauss 3091 Nov 30 17:34 Dockerfile
-rwxr-xr-x 1 opengauss opengauss 657 Nov 30 17:34 entrypoint.sh
-rw-rw-r-- 1 opengauss opengauss 139 Nov 30 18:47 sum.wasm
-rw-rw-r-- 1 opengauss opengauss 68 Nov 30 18:47 sum.rs
-rw-rw-r-- 1 opengauss opengauss 508 Nov 30 18:47 gcd.wat
-rw-rw-r-- 1 opengauss opengauss 727 Nov 30 18:47 fib.wat
drwxrwxr-x 1 opengauss opengauss 4096 Dec 5 14:19 opengauss
[opengauss@489620fe2d9c ~]$
```

```
openGauss=# select wasm_new_instance('/home/opengauss/sum.wasm', 'ws');
wasm_new_instance
-----
2785875771
(1 row)

openGauss=# select wasm_new_instance_wat('/home/opengauss/fib.wat', 'wt');
wasm_new_instance_wat
-----
1940063331
(1 row)

openGauss=# select wasm_new_instance_wat('/home/opengauss/gcd.wat', 'wt');
wasm_new_instance_wat
-----
3780612139
(1 row)
```

```
openGauss=# select ws_sum(125, 251);
ws_sum
-----
376
(1 row)
```

```
openGauss=# select wt_gcd(12345, 345);
wt_gcd
-----
15
(1 row)
```

```
openGauss=#
openGauss=# select * from wasm.instances;
id | wasm_file
-----+-----
2785875771 | /home/opengauss/sum.wasm
1940063331 | /home/opengauss/fib.wat
3780612139 | /home/opengauss/gcd.wat
(3 rows)

openGauss=# select * from wasm.exported_functions;
instanceid | namespace | funcname | inputs | outputs
-----+-----+-----+-----+-----
2785875771 | ws | sum | integer,integer | integer
1940063331 | wt | fib | bigint | bigint
3780612139 | wt | gcd | integer,integer | integer
(3 rows)
```

Benchmark	Runtime	Time (ms)	Ratio
Fibonacci (n = 50)	openGauss-wasm-executor	0.765	1x
	PL/pgSQL	1.714	2x
Fibonacci (n = 500)	openGauss-wasm-executor	0.794	1x
	PL/pgSQL	9.746	12x
Fibonacci (n = 5000)	openGauss-wasm-executor	0.820	1x
	PL/pgSQL	92.720	113x

- 在计算密集型场景下，Wasm执行性能远优于pl/pgsql

openGauss 社区交流

我们希望与各位一起：

共同打造更优秀的openGauss数据库！

共同构筑良好的openGauss数据库生态！



微信公众号



添加微信小助手
即可加入官方社群交流

官网：<https://opengauss.org>

邮箱：contact@opengauss.org

Thanks!