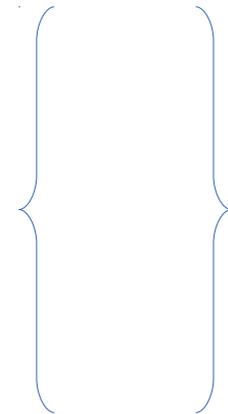
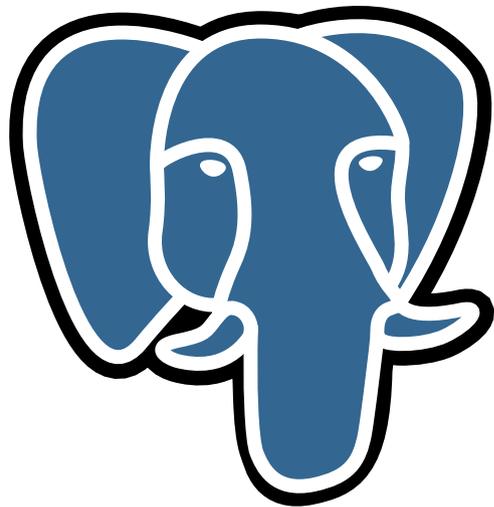


Presenter: Chapman Flack
(chap@anastigmatix.net)
since about 2016, maintainer of



yeah, there's a story

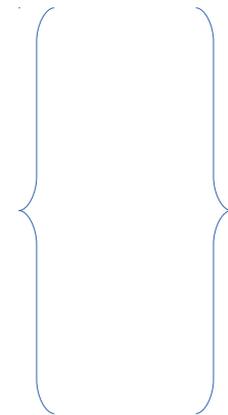
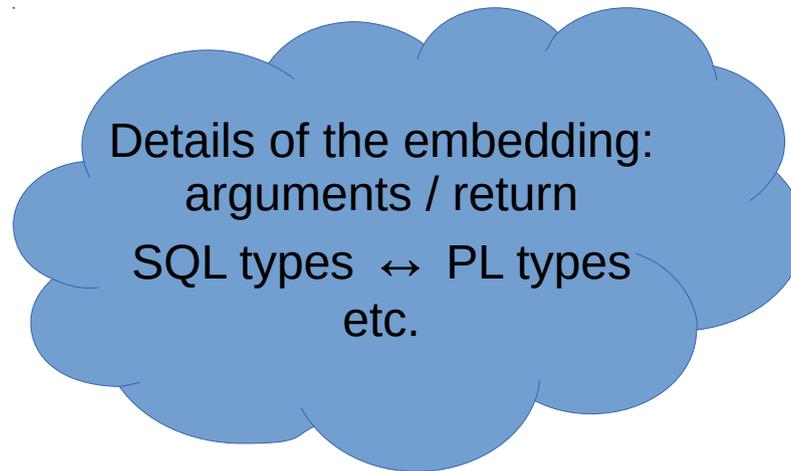
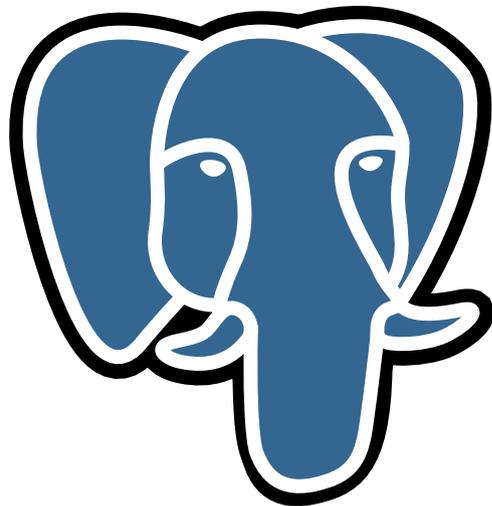
Constraints on a PL



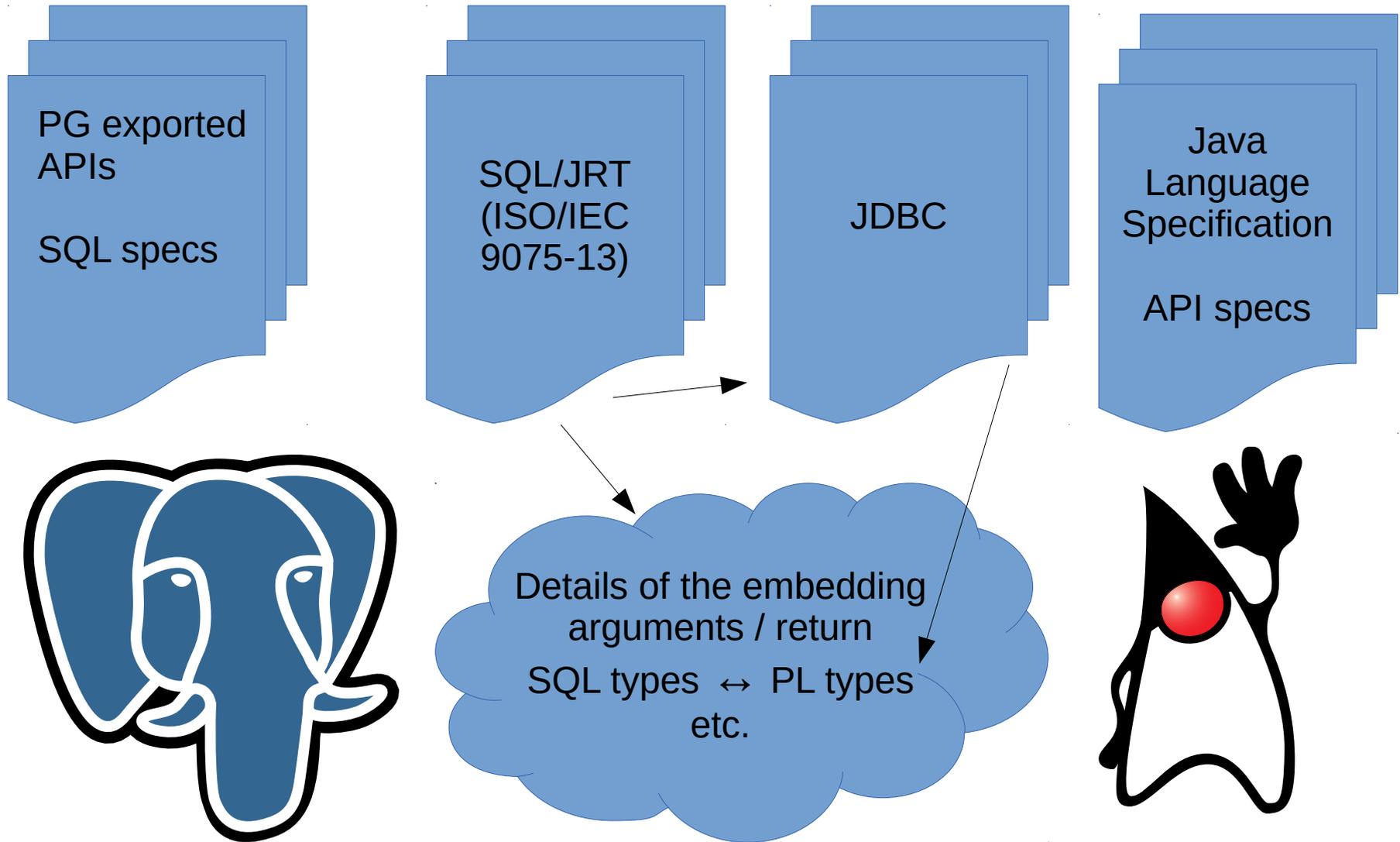
Constraints on a PL



Up to the PL implementor



Constraints on PL/Java

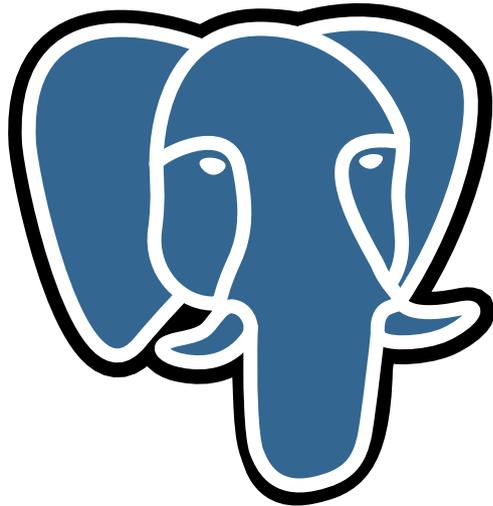


GraalVM™

The TM means it's a trademark, and it ain't mine.

- Oracle Labs, ETH Zurich
- Release numbering starts with 19.0.0 on 19 May 2019! 
- “Community edition” GPLv2 with “Classpath” exception
- www.graalvm.org
- “Polyglot” VM: Java, JavaScript, R, Python, Ruby, LLVM bitcode (from C, ...), other JVM languages (Clojure, Groovy, Kotlin, Scala, ...)
- *“That library is not available in my language. I need to rewrite it.”*
- *“That language would be the perfect fit for my problem, but we cannot run it in our environment.”*
- *“That problem is already solved in my language, but the language is too slow.”* (these three bullets from graalvm site)
- Minimize cost of interlanguage calls **and data accesses**. (DLS 2015, pp. 78-90, doi>10.1145/2936313.2816714)

```
SET pljava.libjvm_location TO  
'/opt/graalvm/jre/.../libjvm.so';
```



- For now: create Java functions in the usual way. They can use `org.graalvm.polyglot` API to run code in other supported languages.
- Not yet: `CREATE LANGUAGE` for other supported languages, `CREATE FUNCTION` in those directly.
- For that, you'll want the more-flexible-datatype-mapping WIP.