



Jan Hofmeier, Kristina Albrecht
TINF15AI-BC
20.04.2018

TensorFlow Architektur

- Was ist TensorFlow
- Anforderungsanalyse
- Einfluss-Faktoren
- Architektursichten:

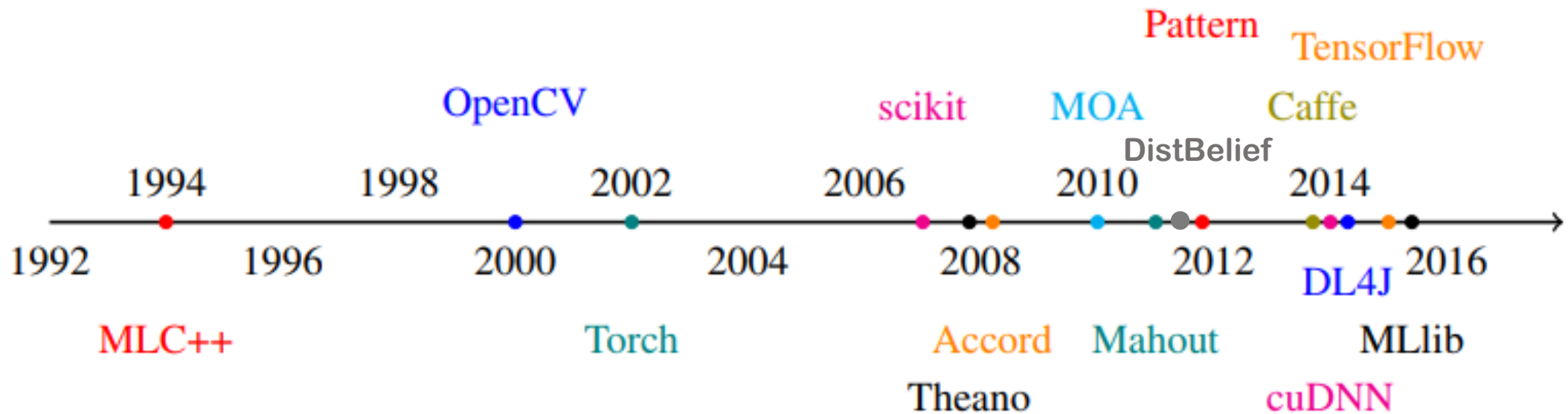
- Szenarien
- Kontextsicht
- Verhaltenssicht
- Struktursicht
- Abbildungssicht

} 1
+
4

- Bewertung der Architektur

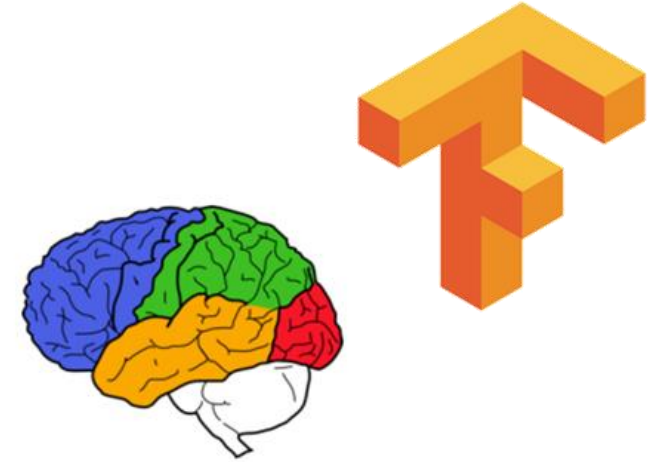


Machine Learning Frameworks



DistBelief

- Vorgänger von TensorFlow
- 2011 von Google Brain entwickelt
- Einschränkungen:
 - Neue Modelle erfordern Änderungen am Core
 - Anpassungen des Parameter-Servers für Gradientenfunktion
 - nur vorwärtsgerichtete Algorithmen (kein Recurrent oder Reinforcement Learning)
 - nur Multi-Core-CPU-Server, keine Unterstützung von GPUs oder anderen Acceleratoren



Anforderungen



ID	Kurzbeschreibung	Anforderung
F1	ML Funktionalitäten	Vektor- bzw Matrizen-Operationen, Lineare Algebra, Statistik, Gradienten etc.
U1	Prototyping	Schnelles Definieren und Testen von Modellen
U2	Produktiver Einsatz	Möglichkeit der Entwicklung komplexer ML Anwendungen
P1	Performance	Effiziente Nutzung verfügbarer Ressourcen
P2	Skalierbarkeit	Umgang mit großen Datenmengen
S1	Portabilität	Portierbar auf verschiedene Systeme
R1	Recovery	Nach einem Abbruch wiederherstellbar

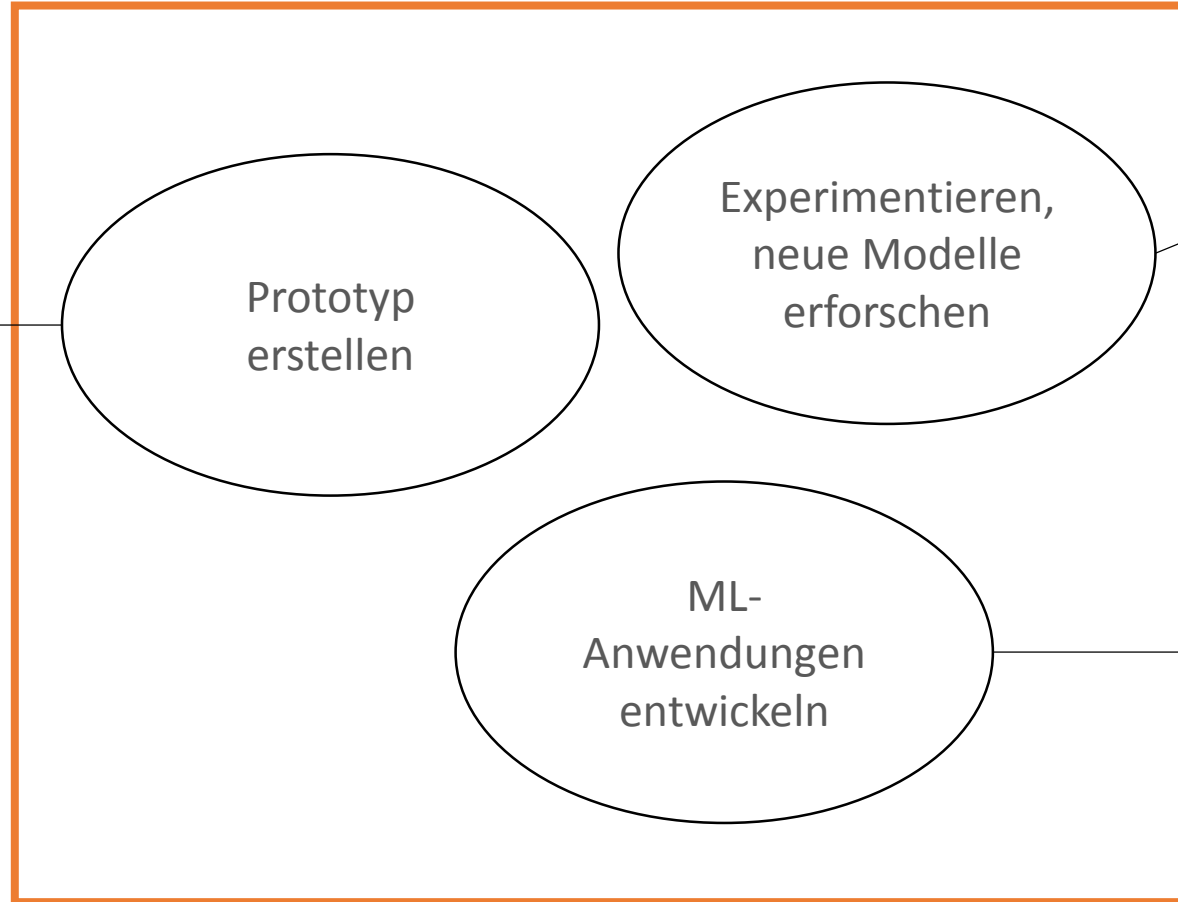
Faktoren



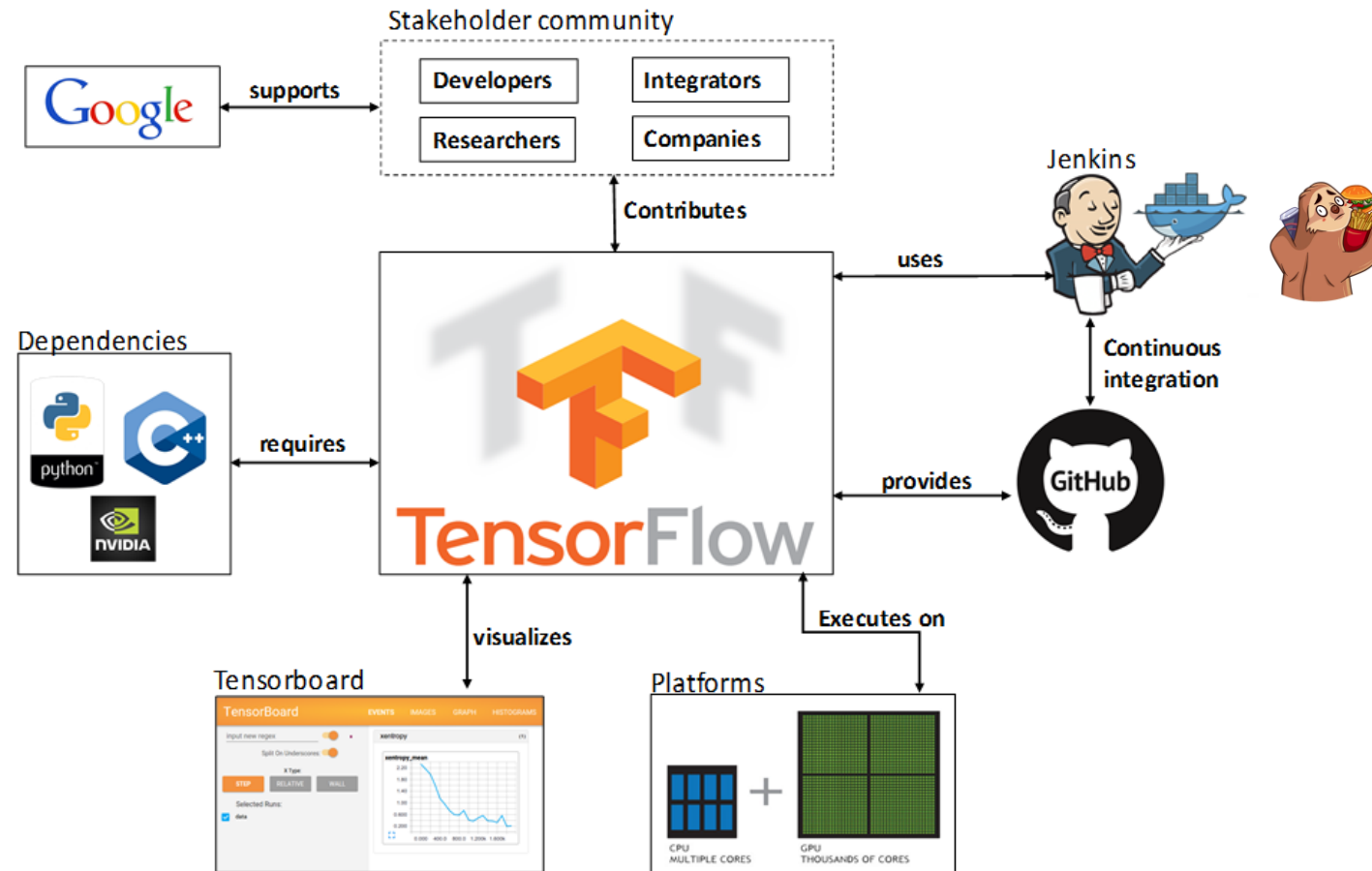
Faktor-Index	Beschreibung	Flexibilität	Einfluss
O1	Community und Contributions	Fest	mittel
O2	Entwickler und interne Nutzer haben Kenntnisse in C++ und Python	Flexibel	mittel
T1	Es werden neue Acceleratoren entwickelt (Bsp.: TPU)	Fest	stark
T2	Die Rechenleistung einer Maschine ist begrenzt, weshalb horizontal skaliert werden muss	Fest	stark
T3	Verschiedene Betriebssysteme (Linux , Mac OS, Windows etc.)	Fest	stark
P1	Neue Bibliotheken können integriert werden	Flexibel	gering
P2	Modelle können sehr komplex werden und viele Daten involvieren	Fest	stark



Szenarien



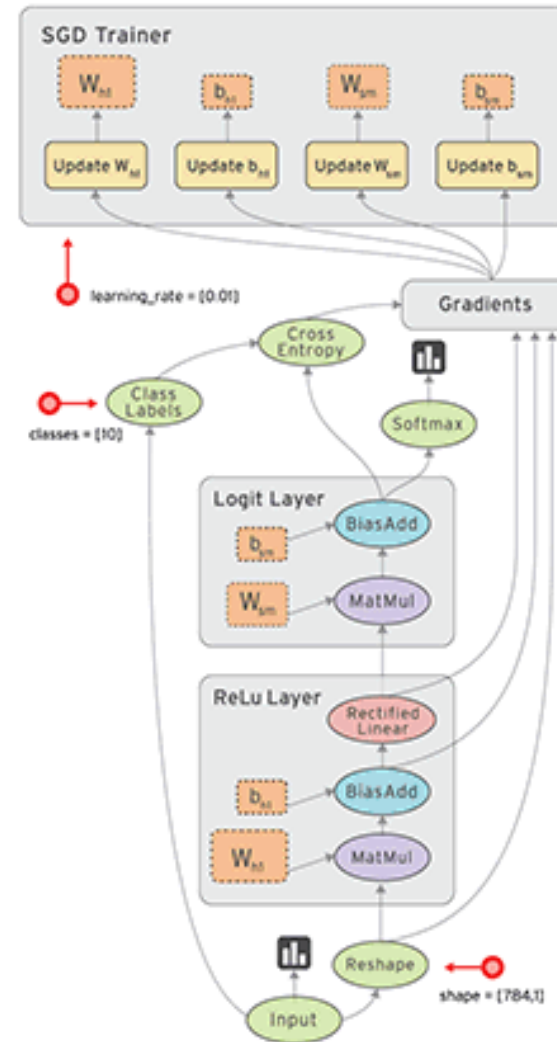
Kontextsicht



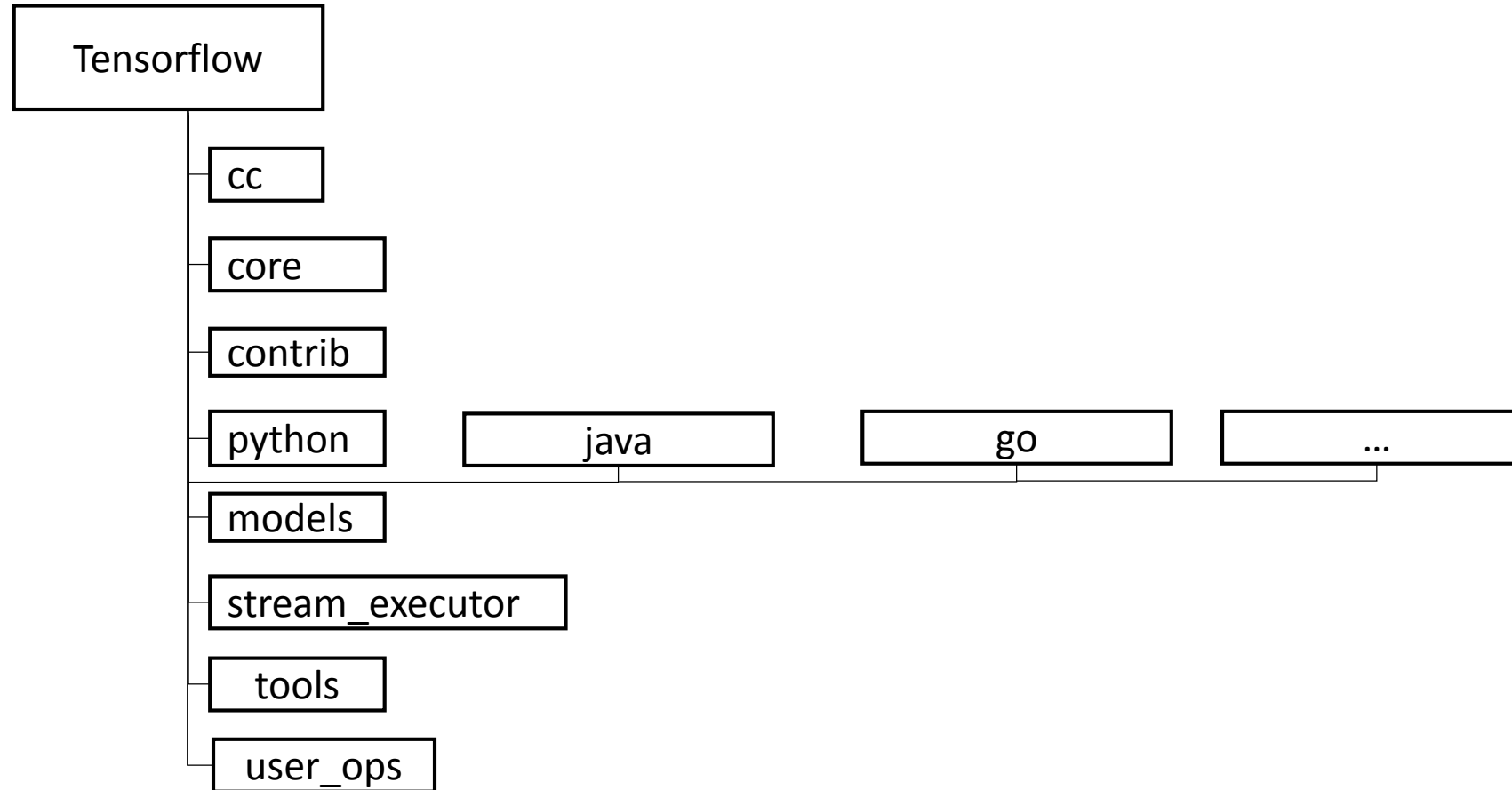
Verhaltenssicht

- Graphs

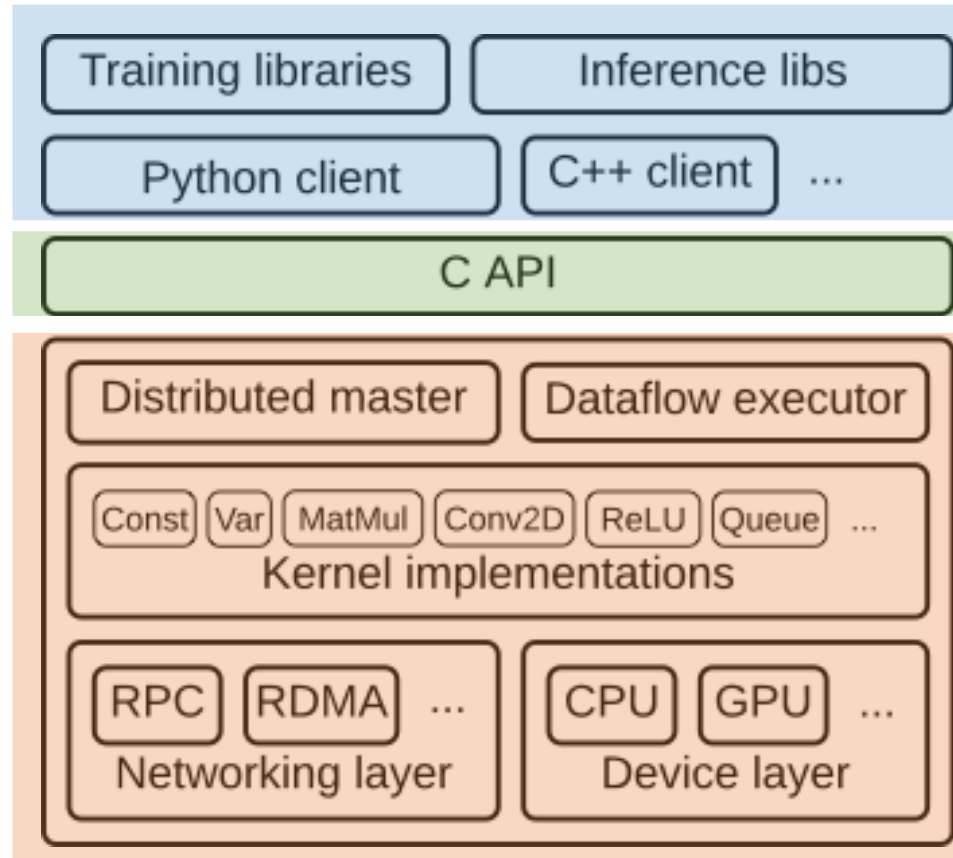
- Gerichtet
- Nodes
- Operators --> MM, add...
- Placeholder --> Eingabewerte
- Variables --> Weights



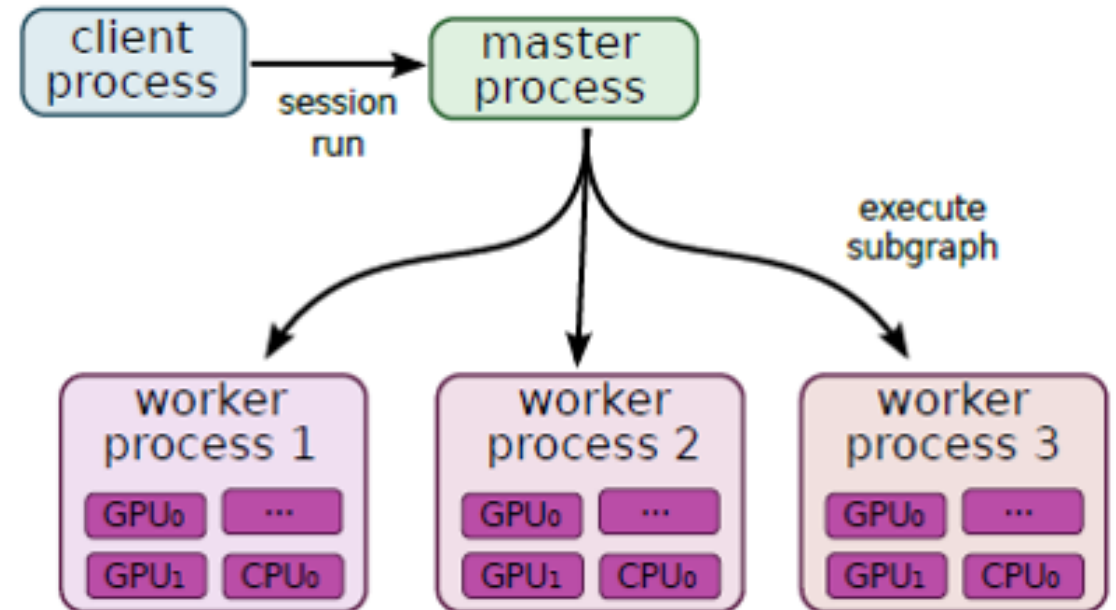
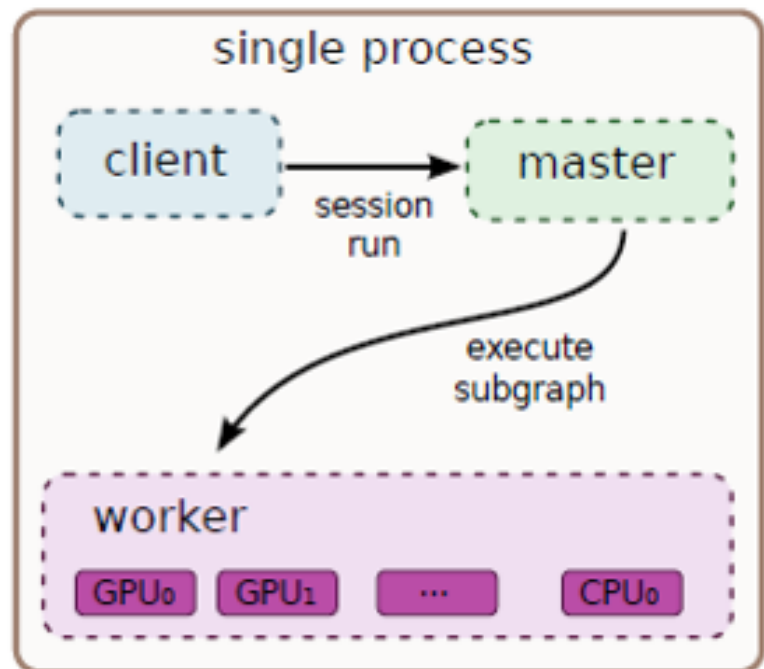
Code Hierarchie



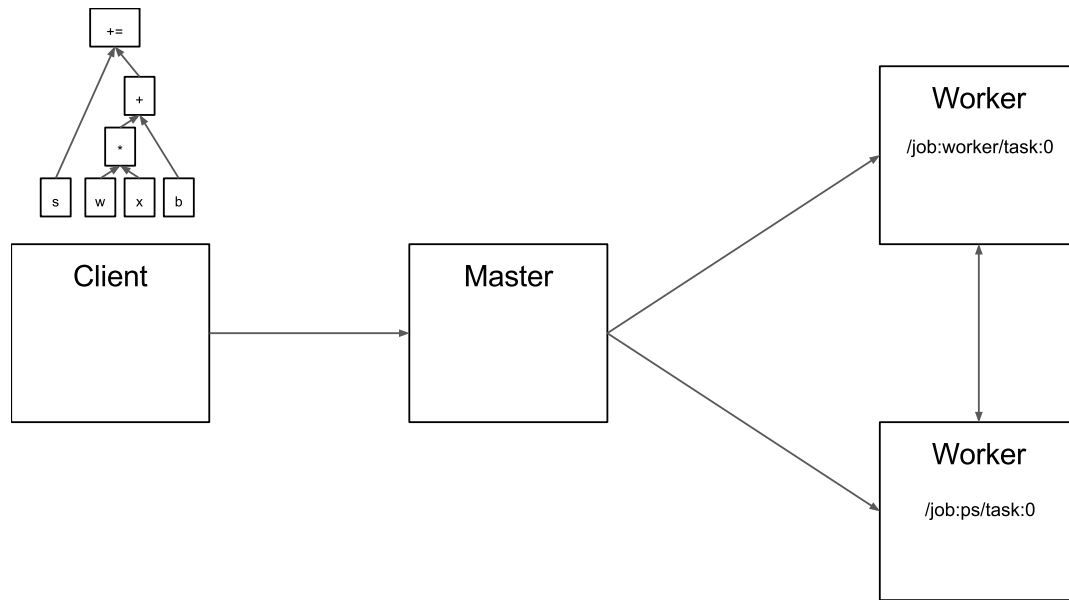
Struktursicht



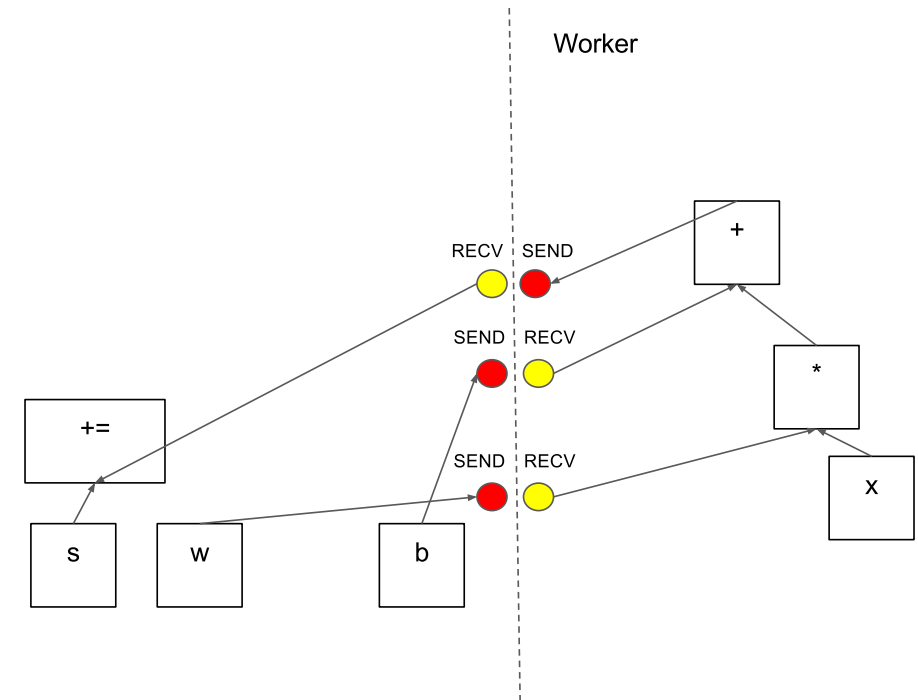
Abbildungssicht



Abbildungssicht



PS



Fazit



ID	Kurzbeschreibung	Umsetzung
F1	ML Funktionalitäten	Externe Bibliotheken, Kernel Implementierungen
U1	Protoyping	High Level APIs
U2	Produktiver Einsatz	Low Level APIs
P1	Performance	C++ Client, Kernel Implementierungen
P2	Skalierbarkeit	Verteilbarkeit
S1	Portabilität	Device Layer
R1	Recovery	Persistierung



