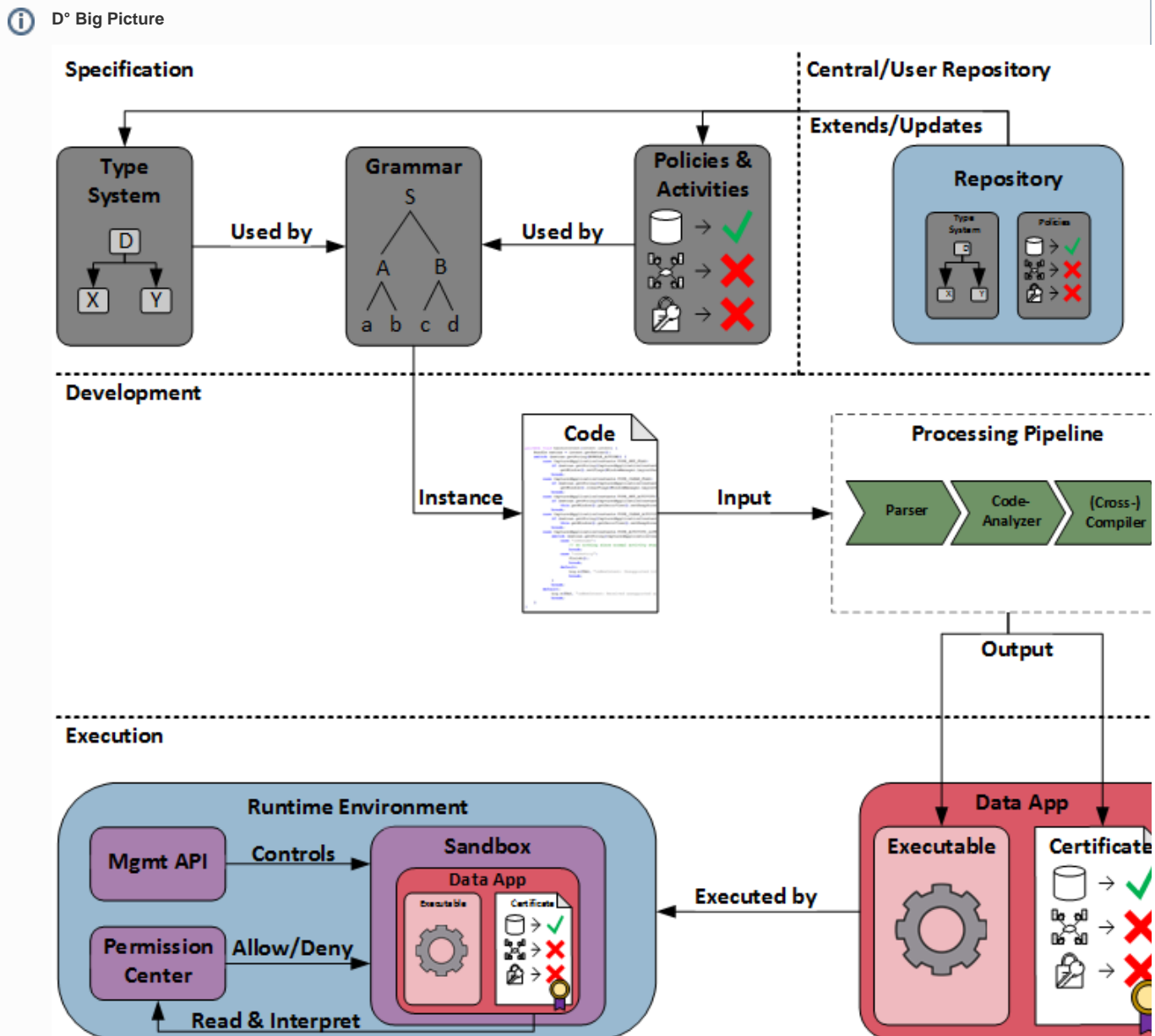


Anforderungsanalyse

Bei der Data App Language handelt es sich um eine Domänenspezifische Programmiersprache, welche den Arbeitstitel D° (gesprochen [digr]) trägt.

Die Programmiersprache zielt auf die Domäne der Datenverarbeitung ab und soll als besonderes Merkmal integrierte Möglichkeiten zur Wahrung der Datensouveränität bieten.



Die Abbildung zeigt die einzelnen Komponenten von D° und deren Zusammenhänge. Dabei lassen sich die Komponenten in drei Ebenen unterteilen, welche nachfolgend kurz erläutert werden: Spezifikation, Entwicklung und Ausführung

Spezifikation

Die Spezifikation der Sprache setzt sich aus mehreren, teils erweiterbaren Modulen zusammen, welche in ihrer Gesamtheit die Programmiersprache D° ausmachen. Das einzige unveränderliche Modul ist das *Grammar*-Modul, welche die Form und Struktur von Applikationen festlegt und somit auch definiert, an welchen Stellen die restlichen Module auf welche Art verwendet werden. Die verbleibenden Module (*Typesystem*, Definitionen von *Activities & Policies*) sind erweiterbar. Zentrale und/oder lokale *Repositories* werden verwendet um die Basismodule mit Erweiterungen zu versehen.

Um Namenskonflikte zwischen verschiedenen Erweiterungen und dem Basismodul zu verhindern werden Namensräume zur eindeutigen Identifikation verwendet.

Das Typsystem erlaubt es, komplexe Abhängigkeiten zwischen einzelnen Datentypen (Komposition, Vererbung) zu definieren. Für jeden Datentyp können direkt im Typsystem Regeln definiert werden, welche verwendet werden um zu validieren, dass Instanzen der Datentypen nur gültige Werte enthalten.

Aktivitäten bestehen aus mehreren Bestandteilen. Neben der eigentlichen Funktionalität sind benötigte Berechtigungen und entstehende Auswirkungen auf die bearbeiteten Daten zu definieren. Beispielsweise verändert die Funktion "anonymisiere" eingehende Daten (Berechtigung) und macht Daten unkenntlich (Auswirkung).

Entwicklung

Der *Code* für eine einzelne Applikation ist eine Instanz der Grammatik, welche die erweiterbaren Module verwendet. Dieser Code ist die Eingabe für die *Processing Pipeline*, welche aus mehreren Schritten besteht und am Ende eine ausführbare *Data App* erzeugt.

Während der Code die verschiedenen Stufen der Processing Pipeline durchläuft werden verschiedene Aktionen durchgeführt.

Durch Analysen des Programmcodes (bspw. Flussanalysen) werden an dieser Stelle bereits die statischen Teile der Policy auf Einhaltung überprüft. Beispielsweise kann überprüft werden, ob auf jedem Ausführungspfad eine bestimmte Aktivität liegt (z.B. "anonymisiere") und somit bestimmte Aussagen über die Verarbeitung der Daten zutreffen (für "anonymisiere": "Daten werden immer unkenntlich gemacht"). Dynamische Anteile im Programmfluss (beispielsweise Schleifen & Sprünge) können diese Auswertung erschweren und/oder unmöglich machen. Diese Teile der Policy werden erst zur Laufzeit überprüft und durchgesetzt.

Der letzte Schritt der Processing Pipeline wird durch einen (*Cross-*) *Compiler* gebildet. Der aktuell entwickelte Cross-Compiler führt eine Übersetzung in Java-Code durch und erlaubt hierdurch die Verwendung von D° auf einer Vielzahl von Geräten und Geräteklassen. Durch den hohen Abstraktionsgrad von D° entstehen bei der Übersetzung in eine andere Programmiersprache komplexe Applikationen in der Zielsprache, welche anschließend in ausführbaren Code übersetzt werden.

Eine direkte Kompilierung in ausführbaren Maschinencode oder andere Programmiersprachen ist zum aktuellen Zeitpunkt nicht geplant.

Zusätzlich zu einer ausführbaren Applikation wird durch die Processing Pipeline ein Zertifikat erstellt, welches diverse Informationen über die Applikation enthält. Neben Informationen, welche zur Integritätsprüfung der ausführbaren Datei dienen, wird beschrieben, welche Teile der Nutzungsbedingungen bereits überprüft wurden und welche zur Laufzeit durchgesetzt werden müssen.

Das Gesamtpaket aus ausführbarer Applikation und Zertifikatsinformation wird im Kontext von D° als *Data App* bezeichnet.

Ausführung

Zur Ausführung der Data Apps wird eine spezielle *Runtime Environment* benötigt, welche die Verwaltung der Data Apps durchführt und Laufzeitpolicies durchsetzt. Ein *Permission Center* liest die Informationen aus den Zertifikaten der Data Apps und entscheidet, ob bestimmte Aktionen bei einer gegebenen Policy ausgeführt werden dürfen. An dieser Stelle werden auch die dynamischen Teile der Policy durchgesetzt, welche nicht zur Übersetzungszeit überprüft werden konnten. Über eine *Management API* ist es dem Anwender möglich Data Apps zu steuern und Einstellungen anzupassen.

Das vorliegende Dokument beinhaltet die Anforderungsanalyse, welche für die weitere Entwicklung von D° verwendet wird. In den folgenden Abschnitten werden Stakeholder und Anforderungen identifiziert und beschrieben.

Stakeholder

Es wurden mehrere Gruppen von Stakeholdern identifiziert, deren Interessen und Ansprüche nachfolgend aufgezeigt werden.

Entwickler (Spracherweiterung)

In diese Gruppen fallen alle Personen, welche D° um zusätzliche Funktionalitäten (bspw. Datentypen oder Aktivitäten) erweitern möchten, um spezielle Anwendungsfälle zu adressieren. Bis zu einem gewissen Grad überschneidet sich diese Gruppe mit der zweiten Gruppe von Stakeholdern - den Entwicklern, welche mit der Sprache entwickeln. Für diese Nutzer ergeben sich die folgenden Ansprüche an die Sprache:

- Definierte Workflows, um die gewünschte Funktionalität zur Sprache hinzuzufügen
- Geringer Overhead bei der Erweiterung der Sprache
- Möglichkeit zur einfachen Verbreitung der eigenen Erweiterungen
- Mechanismen zum Testen eigener Erweiterungen

Neben den Entwicklern, welche die Sprache um Funktionalitäten erweitern, gibt es auch Personen, welche den Cross-Compiler von D° um eine zusätzliche Zielsprache erweitern möchten. Für diese Gruppe ergeben sich die folgenden Anforderungen:

- Definierte Schnittstellen, welche eingehalten werden müssen, um eine korrekte Ausführung der Data Apps zu gewährleisten
- Freiheitsgrade bei der Übersetzung der Zielsprache um Besonderheiten selbiger berücksichtigen zu können

Entwickler (Sprachnutzung)

Alle Personen, welche die Sprache zur Entwicklung von Data Apps (Applikationen, welche mit D° entwickelt wurden) verwenden fallen in diese Gruppe. In dieser Gruppe kann es Überschneidungen mit den Entwicklern, welche die Sprache erweitern, geben. Um diese Gruppe zur effizienten Verwendung von D° zu befähigen, ergeben sich die folgenden Anforderungen:

- Einfache/Bekannte Syntaxkonstrukte (aus klassischen Programmiersprachen) zum einfachen Erlernen der Sprache
- Werkzeugunterstützung während der Entwicklung (Profiler, Debugger, Autovervollständigung, ...)
- Hilfreiche Informationen bei (bspw. Syntax-) Fehlern
- Unterstützung beim Management von Abhängigkeiten
- Simpel ausführbarer Übersetzungsvorgang

Repositorybetreiber

Neben den zentralen Repositories, die von den Nutzern verwendet werden können, um Erweiterungen für die Sprache zu beziehen, können weitere Akteure (private) Repositories betreiben, um eigene Erweiterungen zu verwalten.

- Einfache Wartbarkeit
- Autorisierung & Authentifizierung
- Versionierung
- Komfortable Verwaltung

Endanwender (Nutzer von Data Apps)

Bei den Personen, welche die entwickelten Data Apps nutzen, muss es sich nicht auch um Entwickler handeln. Die Gruppe ist zumindest in Teilen disjunkt mit den oben genannten Entwicklern. Für diese Personengruppe ergeben sich die folgenden Anforderungen:

- Durchsetzung der definierten Policies ist sichergestellt
- Vertrauen in die korrekte Funktionalität der verwendeten Activities
- Möglichkeiten zur Anpassung von Policies zur Laufzeit
- Erhalten von Benachrichtigung bei Policyverstößen
- Kein tieferes Wissen über D° notwendig, um Data Apps verwenden zu können
- Einfaches Management laufender Applikationen

Anforderungen

	Kategorie	Name	Beschreibung
1	Entwicklung (Spracherweiterung)	Definierte Workflows zur Erweiterung der Sprache	Bei der Entwicklung einer Erweiterung für die Sprache müssen die einzuhaltenden Schnittstellen und Rahmenbedingungen bekannt sein, um eine korrekte Funktionalität zu erlauben.
2		Vermeidung von Overhead	Für Spracherweiterungen müssen verschiedene Teile definiert/implementiert werden: Eine Repräsentation in D°, sowie entsprechende Umsetzungen in der Zielsprache(n). Darüber hinaus sollte möglichst wenig zusätzlicher Aufwand anfallen.
3		Testen von Erweiterungen	Während der Entwicklung von Erweiterungen von D° ist es notwendig durch Tests sicherzustellen, dass die gewünschte Funktionalität korrekt bereitgestellt wird. Um diesen Testvorgang zu unterstützen, werden Mechanismen benötigt, welche die Eigenschaften von D° miteinbeziehen.
4		Definierte Schnittstellen zur Ausführungsumgebung	Wenn Entwickler D° für eine neue Zielsprache umsetzen möchten ist es notwendig, dass bestimmte Schnittstellen und Prozesse eingehalten werden, um eine Verwendung in der Laufzeitumgebung zu ermöglichen. Alle hierfür notwendigen Schritte müssen verfügbar und klar benannt sein.
5		Freiheitsgrade des Crosscompilers	Abseits von einzuhaltenden Schnittstellen und Prozessen sollen Entwickler, die D° für eine neue Zielsprache umsetzen, frei in der Ausgestaltung des Programmcodes für die Zielsprache sein. Hierdurch ist es möglich, eventuell vorhandene Besonderheiten/Einschränkungen der jeweiligen Zielsprache zu berücksichtigen.
6	Entwicklung (Data Apps)	Aussagekräftige Fehlermeldungen	Wird während der Implementierung ein (beispielsweise syntaktischer) Fehler gemacht, soll der Entwickler möglichst aussagekräftige Fehlermeldungen erhalten, um den Fehler zu beheben.
7		Werkzeugunterstützung	Um die Entwicklung von Data Apps zu unterstützen sollen Werkzeuge, die aus der klassischen Softwareentwicklung bekannt sind, für D° verfügbar sein.
8		Einfache Syntax	D° sollte auf exotische Syntaxkonstruktionen verzichten und sich stattdessen an bekannten Notationen und Konstrukten aus anderen Sprachen bedienen. Hierdurch wird der Aufwand zum Erlernen von D° gesenkt.
9		Abhängigkeitsmanagement	Es sollte für den Entwickler ausreichen, wenn er die Abhängigkeiten benennt. Der Bezug soll automatisch (spätestens) während der Übersetzung stattfinden. Sollte es sich um spezielle, eigene Erweiterungen handeln, kann es notwendig sein, dass zusätzlich der Bezugsort einer Erweiterung angegeben wird.
10		Übersetzung in die Zielsprache	Bei der Übersetzung einer Data App von D° in eine Zielsprache handelt es sich um einen komplexen Prozess, welcher mehrere Phasen umfasst. Trotz der Komplexität soll es für den Entwickler einfach sein, die Übersetzung seiner Data App auszuführen.
11	Deployment (Spracherweiterungen)	Versionierung von Spracherweiterungen	Aus Kompatibilitätsgründen ist es notwendig, dass verschiedene Versionen einer einzelnen Erweiterung zur Verfügung gestellt werden können. Der Entwickler kann dann durch die Angabe einer Versionsnummer eine bestimmte Version in seiner Data App verwenden.

12		Verwaltung von Spracherweiterungen	Entwicklern muss es möglich sein ihre eigenen Erweiterungen für D° im Repository zu verwalten. Hierfür sind mindestens klassische CRUD-Operationen notwendig.
13		Wartbarkeit der Repositories	Um die Repositories stets auf dem aktuellen Stand zu halten, ist ein Mechanismus, welche eine einfache Wartung erlaubt, wünschenswert. Beispielsweise kann das Repository als fertiger Application Container zur Verfügung gestellt werden, um auf der entsprechenden Infrastruktur installiert zu werden.
14		Nutzermanagement im Repository	Der Zugang zu den im Repository abgelegten Erweiterungen muss nicht zwingend öffentlich sein. Aus diesem Grund ist ein Nutzer- und Rechtemanagement notwendig.
15	Laufzeit	Durchsetzung von Policies	Definierte Policies müssen zur Laufzeit durchgesetzt werden. Dabei soll der Nutzer im Falle von eventuellen Verstößen informiert werden. Darüber hinaus soll der Nutzer einen Überblick über die definierten Policies haben.
16		Management laufender Data Apps	Unter Umständen führt ein Nutzer mehrere Data Apps parallel aus. Daher werden Funktionen zum Management von Data Apps (starten, stoppen, ...) benötigt, sowie eine Übersicht über ausgeführte Data Apps und relevante Informationen über diese Data Apps (beispielsweise die definierten Policies).
17		Einfache Ausführbarkeit	Wenn ein Nutzer eine Data App und die dazu passende Ausführungsumgebung hat, soll die Ausführbarkeit (nach eventuell notwendiger Konfiguration) einfach möglich sein. Es soll nicht notwendig sein tieferes Wissen über D° zu haben, bevor Data Apps ausgeführt werden können.