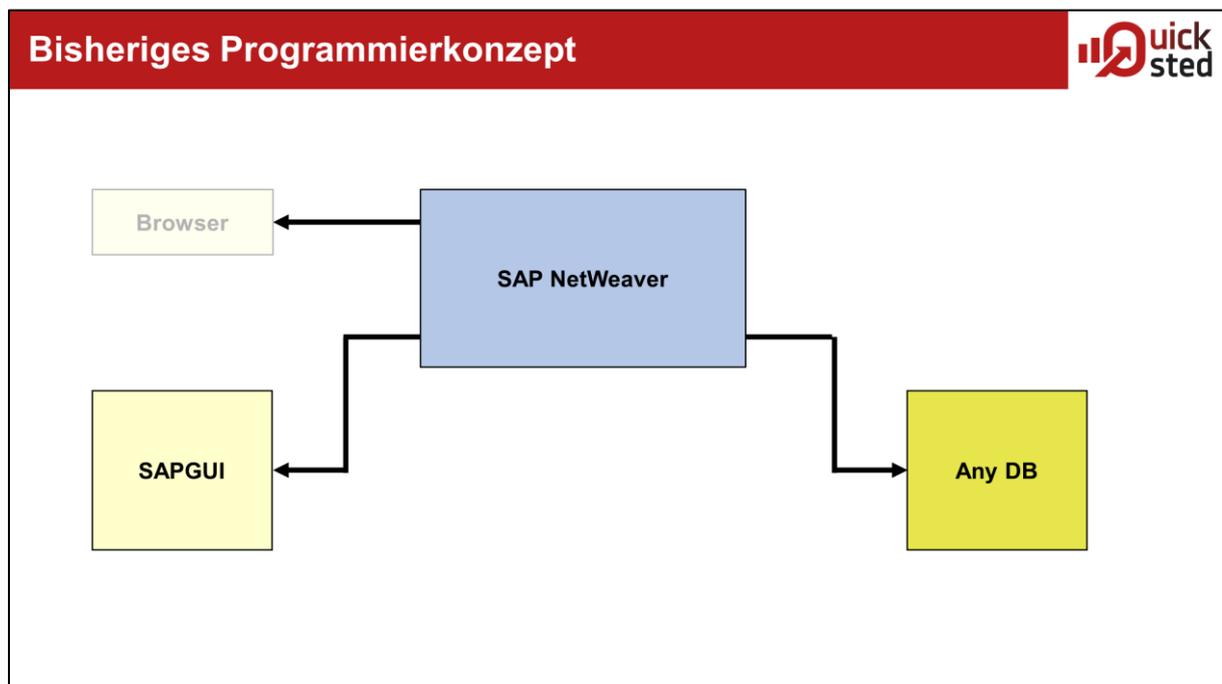


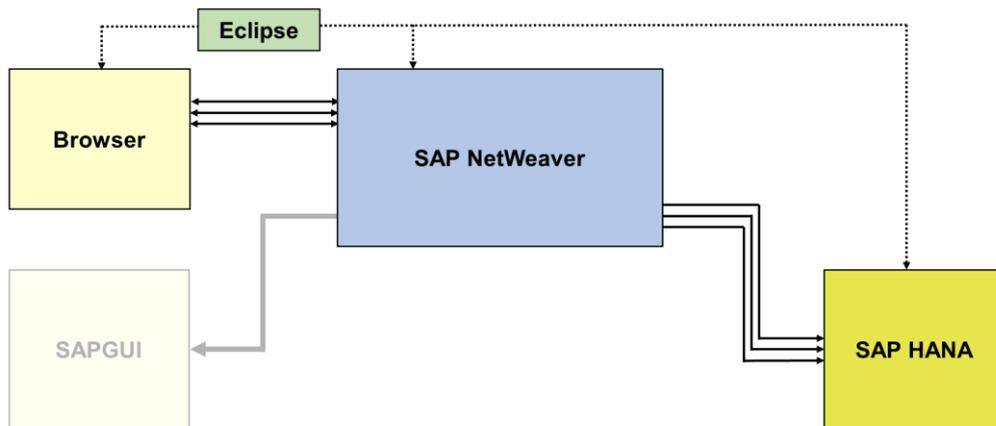
Weiterbildung im Rahmen von SAPs neuem Programmierkonzept

SAPs neues Programmierkonzept

Die bisherige Programmierung bei SAP war im wesentlichen ABAP-zentriert: Die (beliebige, vom SAP NetWeaver unterstützte) Datenbank dient lediglich als Datencontainer, Logik wurde nur in sehr begrenztem Maße (Datenbank-Views, Aggregationen, ...) auf die Datenbank verlagert. Das Frontend ist eine proprietäre Software (SAPGUI), die ebenfalls komplett im SAP NetWeaver (mit ABAP-Mitteln) erstellt wird. Logik ist auch hier nicht vertreten, der SAPGUI wird lediglich zum Rendering verwendet. Es gibt (nicht standardmäßig genutzte) Techniken, die einen Browser als Frontend verwenden (WebDynpro für ABAP [WDA] oder Business Server Pages [BSP]), aber auch diese erstellen das Layout einer darzustellenden Oberfläche komplett im SAP NetWeaver und übergeben dem Browser lediglich einen HTTP-Datenstrom zum Rendern.



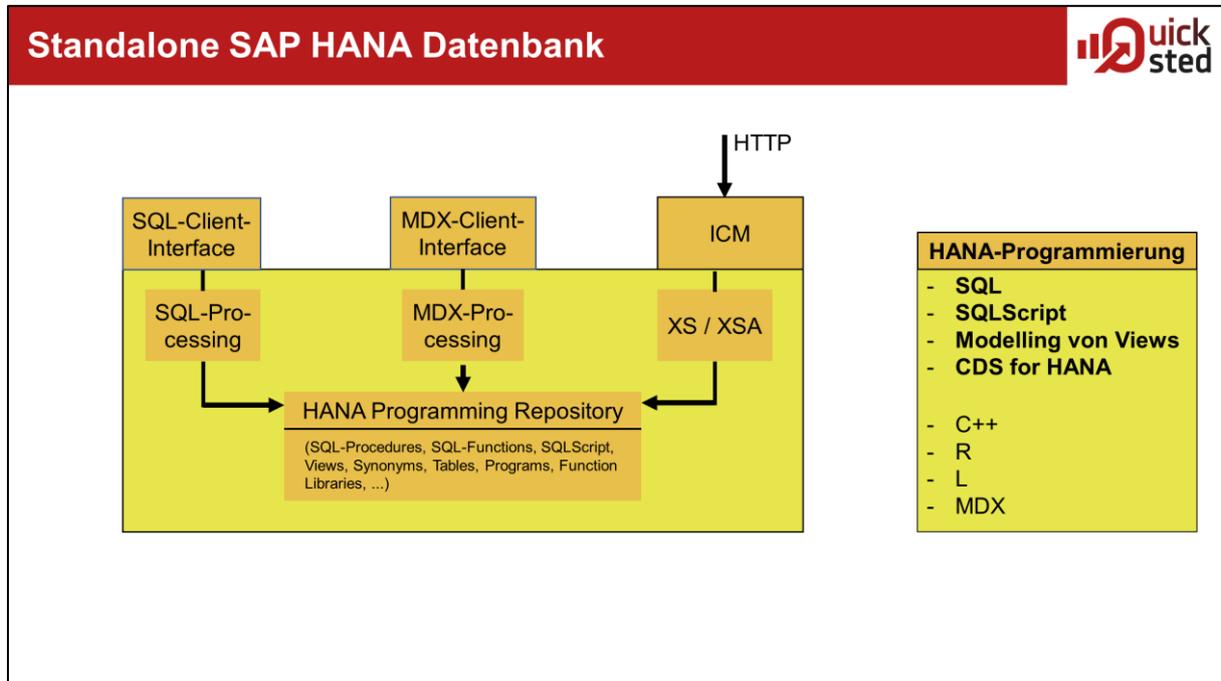
Mit der Einführung von SAP HANA und der Anpassung des Programmierkonzeptes an die neue SAP-eigene Datenbank hat sich die Programmierung für und in SAP drastisch verändert. Zum einen ist die Logik einer Applikation nicht mehr allein auf den ABAP-Teil im SAP NetWeaver konzentriert, sondern kann - wenn der Entwickler dies wünscht - in bedeutendem Umfang sowohl in die Datenbank als auch in das Frontend verlegt werden. Zum anderen hat sich die Anzahl der Möglichkeiten, zwischen Frontend und Applikation bzw. zwischen Applikation und Datenbank zu kommunizieren ebenfalls vergrößert. Es gibt im neuen Programmierkonzept eine Vielzahl von Möglichkeiten, Daten und Informationen zwischen den beteiligten Komponenten auszutauschen. Zusätzlich hat SAP mit Eclipse eine sich außerhalb des SAP-Systems befindliche Entwicklungsplattform integriert.



Für einen Entwickler bedeutet dies, daß er sich auf das neue Programmierkonzept einlassen muß. Das bedeutet nicht, daß seine bisherigen ABAP-Kenntnisse nutzlos geworden sind. Im Gegenteil: Die neuen Programmiertechniken bauen - zumindest im ABAP-Teil - auf den bisherigen ABAP-Releases auf. Dazu kommen allerdings Techniken und Programmiersprachen, die der Verlagerung der Logik in die anderen Komponenten eines SAP-Systems Rechnung tragen müssen.

Datenbank

Architektur, Aufbau und Fähigkeiten einer SAP HANA Datenbank werden an anderer Stelle beschrieben. Soll - zunächst einmal unabhängig von jeglicher Einbindung der Datenbank in ein z.B. SAP-ERP-System - direkt in der Datenbank programmiert werden, sind datenbankspezifische Kenntnisse notwendig.



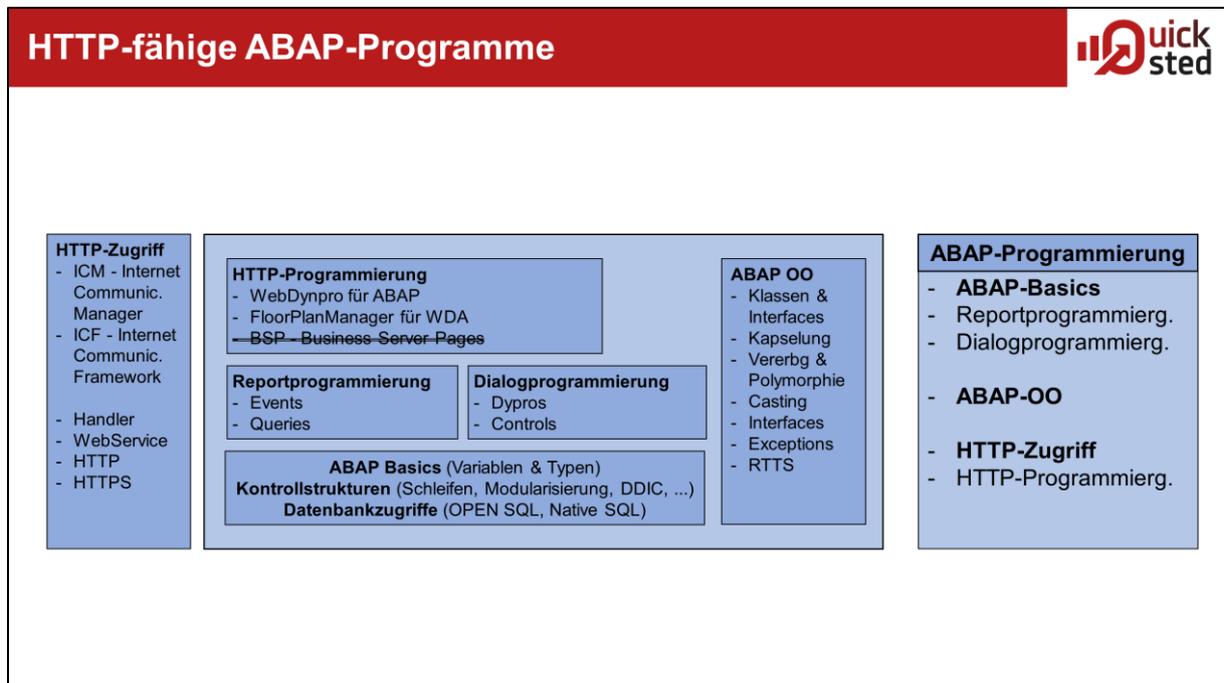
Der Zugriff auf die Daten einer nativen SAP HANA-Datenbank erfolgt in der Regel über den Internet Communication Manager [ICM]. Von hier werden Programme über den Extended Application Server [XS] bzw. den Extended Application Server Advanced [XSA] angesprochen. In der XS können diese Programme in NodeJS (letztlich serverseitigem JavaScript) programmiert werden, die neuere XSA erlaubt sogar Entwicklungen in C++ und Java. Generelle Idee ist es, die XS-/XSA-Applikationen auf HANA-interne Datenquellen zugreifen zu lassen.

Als interne Datenquellen gelten dabei eine Vielzahl von Dingen: Neben den Standard SQL-Befehlen (der ANSI-Standard **SQL-92** ist zu fast 100% umgesetzt), hat SAP den SQL-Standardbefehlssatz mit einer eigenen SQL-Programmiersprache namens **SQL-Script** erweitert. In SQLScript programmierte Prozeduren gelten ebenso als Datenquellen wie Datenbanktabellen oder -views. Zudem ist es möglich durch ein (Eclipse-) Tool ohne Programmierkenntnisse sehr komplexe Views allein auf graphischem Wege zu modellieren. Eine weitere wichtige Datenquelle sind **Core Data Services**, die Datenbank-Views mit hunderten von sogenannten Annotationen anreichern können, die wiederum (als Metadaten übertragen) direkten Einfluß auf das Verhalten von Datenkonsumenten - beispielsweise in der Darstellung - haben können).

Um als Entwickler hier für die Zukunft gerüstet zu sein, sind Kenntnisse im (neuen) SQL und der Programmiersprache SQLScript notwendig (siehe auch unten!). Und es schadet sicherlich nicht, von den Möglichkeiten der View-Modellierung innerhalb der HANA-Datenbank als auch von HANA-CDS etwas gehört zu haben.

SAP NetWeaver / ABAP Platform

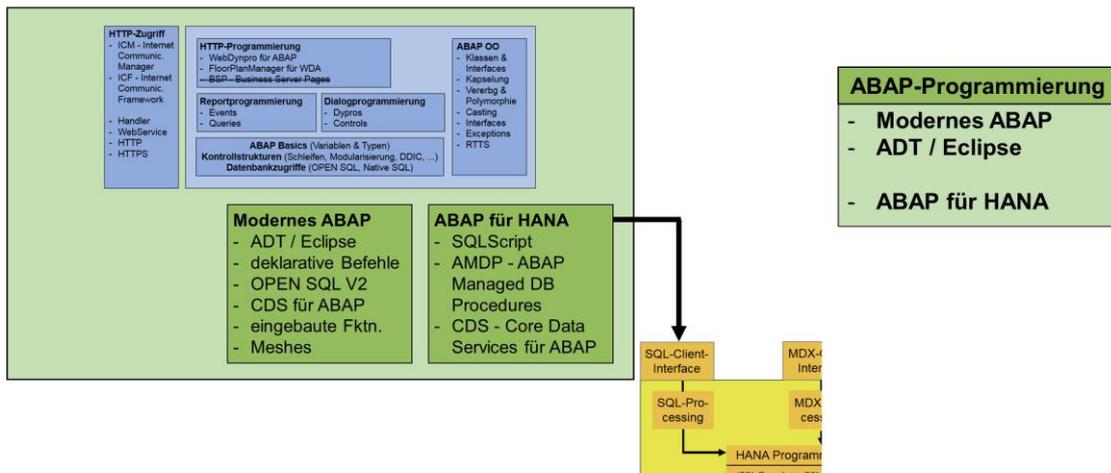
Auf seiten des ABAP-Stacks eines SAP NetWeavers (oder - wie es neuerdings heißt - in der ABAP-Plattform eines SAP-Systems) wird nach wie vor in der Programmiersprache ABAP entwickelt. Diese ist und bleibt die Basis aller Applikationen. Allerdings entstand das Grundgerüst der Sprache bereits vor 20 Jahren und wurde seitdem lediglich "behutsam" angepaßt.



Grundkenntnisse der ABAP-Sprache seien bei einem ABAP-Entwickler einmal vorausgesetzt. Explizit soll jedoch erwähnt werden, daß im modernen ABAP die Objektorientierung eine ungleich größere Rolle spielt als bereits bisher. Die Erfahrung hat gezeigt, daß nicht wenige Entwickler hier Defizite haben. Insofern sei an dieser Stelle explizit auf die Bedeutung von **ABAP-OO** hingewiesen.

Wie oben bereits beschrieben, kann man auch über das HTTP-Protokoll auf ABAP-Coding zugreifen (WDA, BSP). Während BSPs im neuen Konzept keinen Platz mehr haben (sie werden durch SAPUI5-Applikationen abgelöst), sind WebDynpros für ABAP nach wie vor eine Option. Als unbedingt notwendig werden sie im neuen Programmierkonzept jedoch nicht mehr angesehen. Was allerdings wichtig ist, ist ein Verständnis für die **HTTP-Zugriffstechniken** auf ein SAP-System.

Basierend auf diesen bekannten Techniken hat SAP dann die Einführung von HANA zum Anlaß genommen, spezielle, nur auf HANA zugeschnittene Zugriffsarten in ABAP zu implementieren. So ist es mit **ABAP Managed Database Procedures [AMDP]** möglich, native HANA-Prozeduren mit SQLScript als "Gastsprache" mit ABAP-Tools zu entwickeln und dann in die SAP HANA Datenbank zu publizieren. Auch das Core Data Services Konzept ist mit einer eigenen Umsetzung als **Core Data Services für ABAP** neu zum ABAP-Sprachumfang dazugekommen. Diese beiden Techniken lassen sich nur mit Hilfe der **ABAP Development Tools**, einem Eclipse PlugIn, implementieren

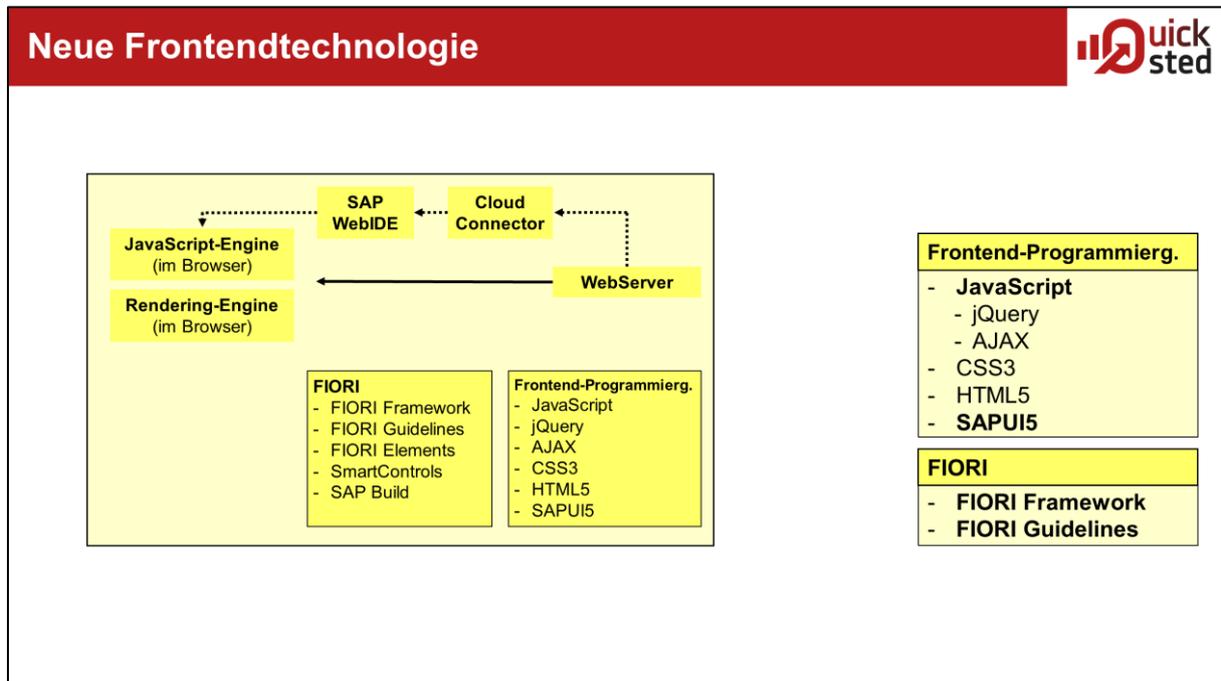


Die Einführung dieser beiden Konzepte beeinflusst auch die grundlegende Basis der allgemeinen Programmiersprache ABAP. Als Konsequenz daraus wurden die grundlegenden Strukturen der Sprache im Release SAP NetWeaver 7.4 im Rahmen von mehreren Support Packages umgestellt. Statt des noch aus R/2-Zeiten stammenden imperativen Befehlsvorrates wurden neue, deklarative Befehlsformen eingeführt (**Ausdrücke, "Expressions"**). Zusammen mit einer ebenfalls runderneuerten OPEN SQL Variante wird ABAP damit auf eine Ebene mit anderen, modernen Sprachen wie C#, Java oder PHP gehoben. Die neuen Ausdrücke erleichtern zudem die Formulierung häufig benutzter Statements und Programmierungstechniken. Im Release SAP NetWeaver 7.5 wird der begonnene Weg konsequent fortgesetzt und den zugehörigen, inzwischen jährlichen Enhancement Packages in die ausgelieferten Softwarekomponenten übernommen.

Da die neuen Entwicklungen der SAP - und dabei insbesondere auch S/4HANA - dieses moderne ABAP verwenden, werden Entwickler um das Kennenlernen und Verstehen des neuen Befehlsatzes nicht herumkommen. Je früher man sich an die neuen Codingregeln gewöhnt, desto einfacher wird einem das Arbeiten im Modernen ABAP fallen.

Frontend

Die neue Frontendtechnologie der SAP setzt auf das HTTP-Protokoll. Tatsächlich verbirgt sich hinter dem Begriff SAPUI5 (oder OPEN UI5 in der freien Variante) die Entwicklung einer reinen JavaScript-WebApplikation, die sich Ihre Daten aus einem SAP-System (oder einem Nicht-SAP-System) über den Aufruf von Webservice besorgt.



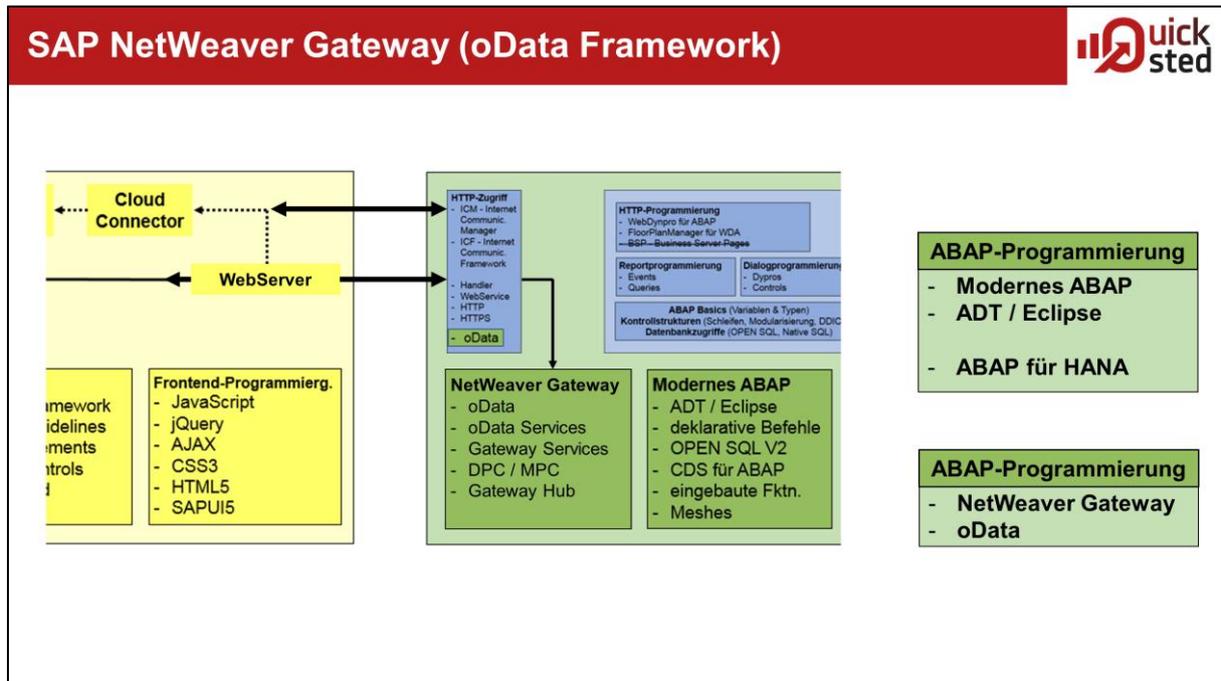
Dieser Ansatz krempelt den von SAP bisher gepflegten Programmierstil komplett um: Das Frontend wird jetzt nicht mehr mit allen benötigten Informationen (passiv) beliefert, sondern verlangt im Gegenteil nun aktiv die Daten. Zudem steuert es, welche Daten gerade notwendig sind. Die Programmierung erfolgt im Wesentlichen über JavaScript. Die dabei eingesetzten JavaScript-Bibliotheken werden als **SAPUI5** zusammengefaßt und beruhen auf der aktuellen Implementierung **HTML5** der Seitenbeschreibungssprache HTML, den Cascading Stylesheets (**CSS3**), Asynchronous JavaScript and XML (**AJAX**) und der JavaScript-Bibliothek **jQuery**.

Um die Verwendung der SAPUI5-Bibliotheken einheitlich zu gestalten, hat SAP darüber hinaus eine Verwendungsrichtlinie namens **SAP Fiori** konzipiert, die genau beschreibt, wie eine SAP Fiori Applikation auszusehen hat, wie die einzelnen Bestandteile der SAPUI5-Bibliotheken zu verwenden bzw. auf einer Weboberfläche anzuordnen sind, und wie sich eine SAPUI5-Applikation im Allgemeinen zu verhalten hat. Es gibt sogar spezielle, SAP Fiori-Anwendungen vorbehaltene UI-Controls.

Auch im Frontend-Umfeld sollte ein SAP-Entwickler zumindest die Grundzüge der SAPUI5-Entwicklung kennen, um zu verstehen, wie Webservice-Daten aufbereitet werden müssen, die von einem auf SAPUI5 beruhenden WebApplikation angefordert werden. Die Kenntnis von SAP FIORI dagegen ist optional, da über dieses Framework lediglich das Aussehen einer WebApplikation gesteuert wird.

Kommunikation Frontend <-> SAP NetWeaver / ABAP Platform

Als letzter Punkt ist noch die Kommunikation zwischen SAPUI5-Frontend und SAP NetWeaver zu betrachten, oder besser gesagt: Die Schnittstelle zwischen den beiden Komponenten. Wie gesagt werden Daten vom Frontend über HTTP-WebServices aus dem SAP NetWeaver angefordert. Für das darin verwendete Protokoll hat SAP allerdings auch eine ausgewiesene Präferenz: Die Daten sollten im **oData-Protokoll** verpackt sein, d.h. der SAP-NetWeaver muß auf oData-Anfragen reagieren und die angeforderten Daten im oData-Protokoll ausliefern können.

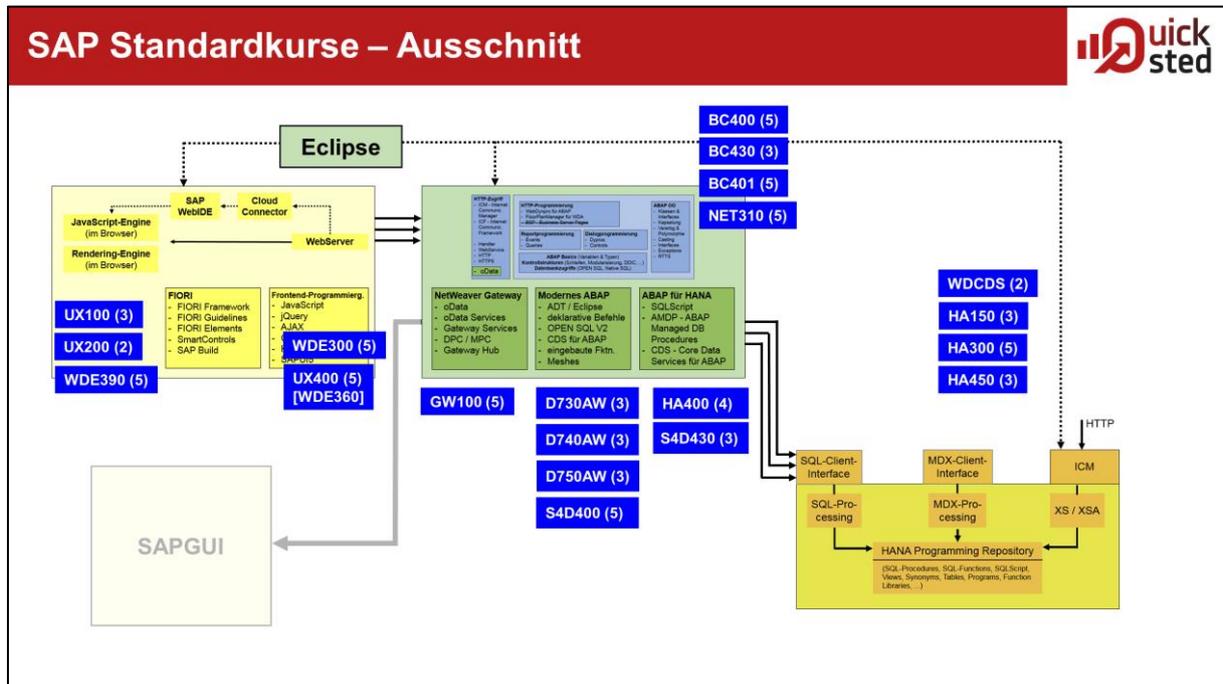


Das dafür im SAP NetWeaver zuständige oData-Framework heißt **SAP NetWeaver Gateway** und ist ebenfalls Bestandteil der unter den Begriff "Modernes ABAP" fallenden Erweiterungen zu SAP NetWeaver 7.4.

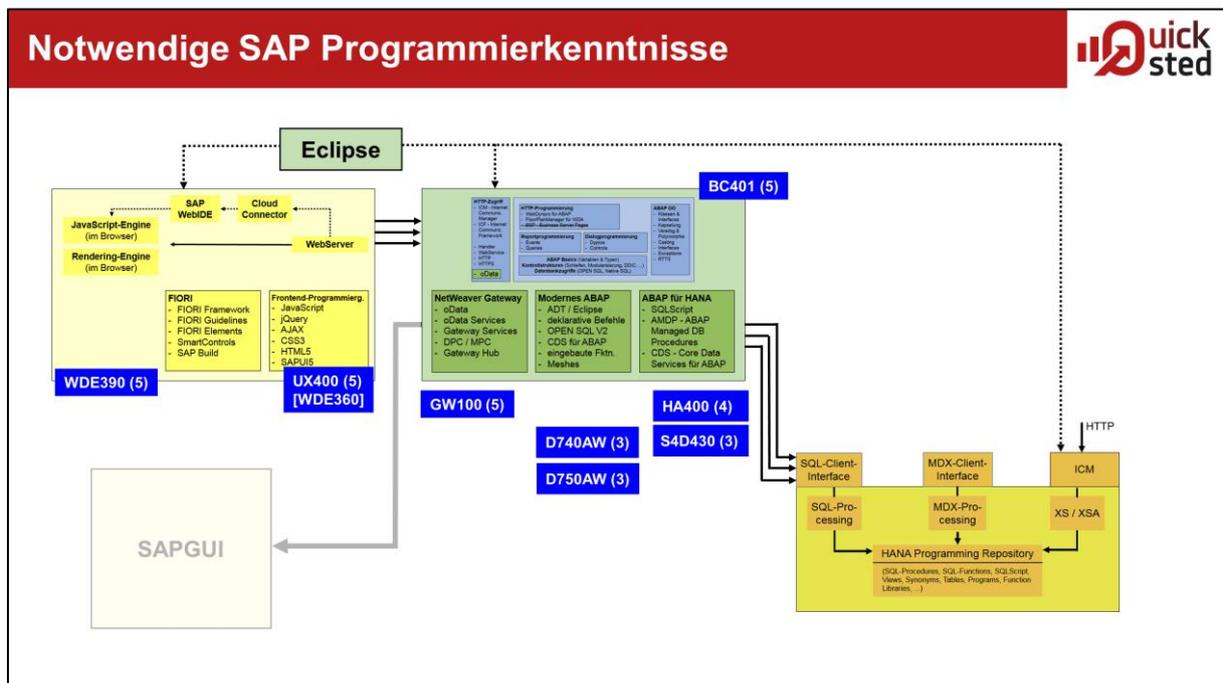
Für den Entwickler sind an dieser Stelle Kenntnisse der Funktionsweise des oData-Protokolls und des SAP NetWeaver Gateways erforderlich.

Lernpfade

In der folgenden Abbildung sind die bisher besprochenen Einzelteile des neuen Programmierkonzeptes sowie eine Reihe von SAP-Schulungen aufgeführt, die sich mit der Wissensvermittlung zu diesen Themen beschäftigen.



Nicht alle der gezeigten Kurse sind notwendig, um die Grundlagen des neuen Konzeptes zu verstehen. Die nächste Abbildung zeigt daher die Kurse, die in diese Basics für die einzelnen Themen einführen.



Abschließend noch als Vorschlag ein Lernpfad, der die unterschiedlichen Themengebiete optimal miteinander verknüpft.

Vorgeschlagener Lernpfad









BC401 (5 Tage)	ABAP Objektorientierung	A03 (68 Quicks)
D740AW (3 Tage)	Modernes ABAP in SAP NetWeaver 7.4 und 7.5	S11 (70 Quicks)*
D750AW (3 Tage)		
HA400 (4 Tage)	ABAP für HANA	S13 (40 Quicks)*
S4D430 (3 Tage)		
UX400 (5 Tage) [WDE360]	SAPUI5	S32 (65 Quicks)
WDE390 (5 Tage)	[optional: SAP FIORI]	S33 (75 Quicks)
GW100 (5 Tage)	SAP NetWeaver Gateway	S34 (55 Quicks)**
		S21 (55 Quicks)

Ein **Quick** ist eine Lerneinheit, bestehend aus einem ca. 15minütigen Video, einer schriftlichen Dokumentation und einer Übung. Die Übung wird ebenfalls von einem Video begleitet, das die Lösungsschritte zeigt.

Weitere Einzelheiten:
www.quicksted.com

*) Realisierung voraussichtlich Q2/2022
 **) Realisierung voraussichtlich Q4/2022

Als Grundlage dient das Lernen bzw. das Auffrischen der Objektorientierung in ABAP. Auf dieser Basis lassen sich die neuen Ideen des Modernen ABAP am einfachsten in das vorhandene Wissen integrieren. Als Erweiterung und Fortführung des neuen Teils der ABAP-Programmiersprache dienen dann die Elemente, die speziell für die SAP HANA Datenbank zu ABAP hinzugekommen sind.

Ein anschließender Blick in die Frontendtechnik SAPUI5 zeigt, wie Daten aus einem SAP-System verwendet bzw. dargestellt werden können. Eine Beschäftigung mit dem SAP FIORI-Konzept ist optional. Und das Kennenlernen des SAP NetWeaver Gateways bzw. des oData-Protokolls verknüpft zum Abschluß die ABAP-Techniken des Backends mit den SAPUI5-Darstellungstechniken im Frontend.