

→ **Checkliste „SaaS Readiness/PaaS Selection“**

Allgemeine Parameter
Welche PaaS-Dienste werden benötigt?
Welche PaaS-Architektur passt am besten zu den Anforderungen des Kunden? Ist die bisherige Technologie zukünftig noch relevant? Welche Rahmenbedingungen spielen ggf. eine Rolle?
Gibt es geografische relevante Aspekte (Internationalität, gesetzliche Einschränkungen des Ortes)?
Spielen Präferenzen der (potenziellen) Kunden eine Rolle?
Haben Sie einen direkten Zugang zu den (potenziellen) Anwendern Ihrer Software?
Inwieweit ist zukünftig ein Auswechseln der PaaS oder wesentlicher Bausteine, z. B. des Messaging oder der Bus einzuplanen?
Wie stark ist die SaaS-Applikation in kundenseitige Prozesse und Infrastrukturen integriert?
Welche Backend-Prozesse (Rechnungsstellung, Bonitätsprüfung, Geldeinzug, ...) werden benötigt?
Wo soll die SaaS-Applikation (physisch) betrieben werden?
Wie hoch ist der spezifische Installations-/Integrationsaufwand?
Sind die Anwendungen bereits über eine moderne Web-Oberfläche nutzbar?

Spezifische Themenfelder (fast immer relevant)
Plattform & Schnittstellen
Ist die PaaS weitgehend plattformunabhängig? Setzt sie eine spezifische Ablaufumgebung voraus? Ist ein spezifisches Architekturmodell bereits weitgehend festgelegt (z. B. force.com)? Ergeben sich daraus deutliche Einschränkungen (dotnet oder Java)?
Welche Tools bzw. Entwicklungsumgebung (z. B. Visual Studio, Eclipse) können unter Berücksichtigung der PaaS genutzt werden?
Sind die von der PaaS angebotenen Interfaces für die Verbindung mit der Anwendung geeignet?
Bietet die PaaS eine umfassende Auswahl von Standard-Komponenten (z. B. Data Acquisition, Import/Export, Extraction, Classification, Rules, Reports, Billing & Payment, PDF Tools) um typische Prozesse abzudecken?
Unterstützt die PaaS die dynamische Buchung von Cloud-Ressourcen (Public oder Private)?
Multimandantenfähigkeit (Multi-Tenancy)
Sind Ihre Lösungen für viele Kunden/Mandanten bereits getrennt einsetzbar?
Basiert Ihre Lösung auf einer modernen einfach skalierbaren Server-Architektur?
Welche Anzahlen von Mandanten werden erwartet, d. h. welcher Grad an Skalierbarkeit wird gefordert?
Ist die Anwendung bereits in einzeln nutzbare Komponenten (Services) aufgeteilt (Separation of Concerns, Single Responsibility Principle)?
Authentication & Identity Management
Bietet die PaaS eine User Administration, Single Sign On (SSO) und mandantenunabhängige Logik für die Verwaltung von Rollen und Rechten?
Können zusätzlich kundenseitige Identity Management Lösungen (z. B. Active Directory, LDAP, OpenID oder Crowd) seitens der PaaS integriert werden?
User Interface (UI)
Benötigt die Anwendung eine eigene Nutzeroberfläche oder läuft sie nicht sichtbar im Kontext einer anderen Anwendung?
Ist die fachliche/technische Funktionalität „sauber“ von der Benutzeroberfläche getrennt?
Müssen UI-Anpassungen im Frontend vorgenommen werden? Können zusätzliche Funktionen ggf. über ein eigenständiges UI bereitgestellt werden?





Spezifische Themenfelder (fast immer relevant)

Sicherheit/Compliance

Werden die Daten, Protokolle und Metadaten in einem System mit mehreren Mandanten sicher abgegrenzt, so dass (auch versehentlich) kein spill-over zu einem anderen Mandanten erfolgen kann?

Spezifische Themenfelder (bei Bedarf)

Billing & Tracking

Erzeugen die Applikations-Komponenten Abrechnungsinformationen? Bietet die PaaS Standard-Tools für Abrechnungsinformationen (z. B. pro Transaktion, lastabhängig oder nutzerabhängig)?

Ist eine einfache Schnittstelle zu einer Buchhaltung vorgesehen?

Skalierbarkeit

Bietet die PaaS ein Konzept oder technische Unterstützung zur Skalierbarkeit der SaaS-Anwendung?

Sind hohe Skalierbarkeit sowie saubere Trennung verschiedener Systembereiche (Separation of Concerns) erforderlich bzw. angestrebt? Würde hierzu eine Bus-Logik z. B. auf Basis eines Enterprise-Service-Bus (ESB) passen?

Interoperabilität

Ist eine Interoperabilität mit Webanwendungen, klassischer on-premise Software oder Integration mit Legacy-Software erforderlich? Sollte diese durch eine Bus-Architektur oder ein PaaS unterstützt werden?

Konfigurierbarkeit/Flexibilität

Sind Abläufe und Geschäftslogik Ihrer Anwendungen konfigurierbar?

Wird das Datenmodell regelmäßig geändert? Gibt es pro Mandant individuelle Anforderungen an das Datenmodell?

Security

Ist die technisch sichere Datenübertragung gewährleistet?

Können die Daten sicher aufbewahrt werden?

Ist die Kommunikation zwischen eigener Infrastruktur und dem Service Provider ebenfalls gesichert?

Maintenance/Updates

Wie erfolgt das Update von Software-Versionen der Anwendung, ohne Beeinträchtigung der Nutzer? Wird dies durch die PaaS unterstützt?

Sind häufig Updates erforderlich? Ist bei höherem Zeitbedarf das System noch zugreifbar?

Monitoring

Gibt es ein „Monitoring“ mit Health-Check zur Überwachung der Funktionstüchtigkeit und des Verhalten der Anwendungen sowie die Einhaltung von vereinbarten Service Levels?

Prozessintegration

Ist eine Integration der SaaS-Anwendung in kundenseitige Prozesse gefordert? Gibt es hierzu Schnittstellen seitens der PaaS? Bietet die PaaS ein eigenes BPM oder eine Integration zu einem BPM-Produkt?

Standards

Sind bestimmte Standards, etwa durch Branchenspezifika oder Compliance-Anforderungen, einzuhalten?

Orchestrierung von Services

Ist eine Orchestrierung von Services erforderlich? Soll diese durch PaaS oder ggf. Individualprogrammierung umgesetzt werden?





Spezifische Themenfelder (bei Bedarf)

Provisionierung

Ist eine weitgehende Automatisierung für die Bereitstellung eines neuen Mandanten erforderlich?
Reichen ggf. mehrere Stunden oder Tage?

Gibt es Geschäftsprozesse, die mit der Mandatenprovisionierung verbunden sind, z. B. eine
Vertragszusendung oder Freischaltungs-E-Mail?



Der Autor des Whitepapers

Bernhard Cygan

Telefon: +49 (0)30 469 99 29 04

E-Mail: bernhard.cygan@stoneone.de

Kontakt:

Bernhard Cygan

Solution Architect

StoneOne AG

Keithstraße 6

D-10787 Berlin

bernhard.cygan@stoneone.de

www.stoneone.de



stone one
The Web Service Factory