

## Vergleich: BOI FreeDa vs. MS Access

Kriterium	BOI FreeDa	Microsoft Access
→ Time-to-Value	Einführung in Wochen möglich; keine Definition von Masken nötig	Erste Masken und Anwendungen schnell erstellt, aber wenig robust bei wachsender Komplexität
→ Technologie	Web-Anwendung mit zentralem Zugriff, jederzeit über Browser nutzbar	Desktop-Datenbank (lokal oder im Netzwerk geteilt)
→ Mehrbenutzerbetrieb	Bewährte Architektur für hohe Benutzerzahlen (100+ gleichzeitig)	Nur eingeschränkt; ideal für Einzelnutzer oder kleine Teams (bis zu 10 gleichzeitige Zugriffe) Performanceprobleme bei vielen gleichzeitigen Usern
→ Skalierbarkeit & Performance	Stabile Architektur für kleine bis große Datenmengen: von 1 bis 10.000+ Tabellen	Für kleine Datenmengen geeignet; bei größerer Datenlast instabil
→ Revisionsicherheit & Compliance	Out of the box: Audit Trail, Versionierung, nachvollziehbare Änderungen inkl. Benutzer-/Zeitstempel	Keine integrierte Versionierung oder Audit Trails; revisions-sichere Erweiterungen müssen selbst programmiert werden
→ Governance & Rollen	Feingranulare Rollen- und Rechteverwaltung	Nutzer- und Rechteverwaltung eingeschränkt; keine granulare Rollensteuerung
→ Pflege-Prozesse	Standardisierte 2-, 4- und 6-Augen-Prozesse inkl. Änderungsansicht für die Kontrolle	Individuelle Formulare; Prüfmechanismen und Freigaben müssen selbst implementiert werden
→ Datenbank-Anbindung	Unterstützt alle gängigen relationalen Datenbanken	MS Access-Datenbanken oft Insellösung; Verbindung zu anderen RDBMS eingeschränkt
→ APIs & Integration	Standardisierte Schnittstellen wie z.B. REST-API für plattformübergreifende automatisierte Prozesse	Schnittstellen nur eingeschränkt und mit Zusatzaufwand nutzbar, für Desktop-Anwendungen konzipiert

## Vergleich: BOI FreeDa vs. MS Access

➔ <b>Betrieb</b>	On-Premise und Cloud-fähig; Docker-basiert	Lokal oder als Datei geteilt; keine professionelle Lifecycle-Steuerung, keine zentrale Wartung
➔ <b>Flexibilität</b>	Hohe Konfigurierbarkeit, Erweiterbarkeit über APIs und standardisierte Module	Maximale Freiheit für kleine Szenarien, aber keine nachhaltige Governance und Skalierbarkeit
➔ <b>Integration</b>	Standardisierte APIs und Schnittstellen für Integration in Geschäftsprozesse	Anbindung an andere Systeme aufwändig; meist über ODBC oder manuelle Prozesse
➔ <b>Benutzerfreundlichkeit</b>	Moderne Web-Oberfläche speziell für Tabellenpflege ohne Datenbankkenntnisse; keine Pflegemasken nötig	Bekannte MS Office-Oberfläche, einfache Maskenerstellung, aber begrenzte Usability und moderne Anforderungen (z. B. Barrierefreiheit) schwer umsetzbar
➔ <b>Sicherheit</b>	Unternehmensweite Sicherheitsmechanismen, Protokollierung und Rechteverwaltung	Lokale Datei- oder Netzwerksicherheit; keine durchgehenden Sicherheitskonzepte