Wie bereits erwähnt ist bei modernen Anwendungen ein Kompromiss zwischen der Daten-Konsistenz und der Verfügbarkeit zu finden. Da moderne Anwendungen einen deutlichen Trend in Richtung Web-Anwendungen zeigen, wird hier versucht diesen Kompromiss am Beispiel einer Web-Anwendung aufzuzeigen. Ein Beispiel für die Auswahl der Konsistenz wäre eine transaktionsorientierte Kontenverwaltung, wobei es zu längeren Wartezeiten kommen kann.

Die andere Möglichkeit wäre die Verfügbarkeit in den Vordergrund zu stellen. Dabei erhält man sehr geringe Antwortzeiten, jedoch kann dies aber auch zu inkonsistenten Daten führen.

Abstrakter formuliert muss man sich die Frage stellen, ob man eine hohe Sicherheit oder eine hohe Lebendigkeit möchte. Wollen wir die Sicherheit einer Konten-Verwaltung einer Bank oder die Lebendigkeit einer Web 2.0-Anwendung?

Für Web-Dienste wie Amazon oder Google steht die Lebendigkeit im Vordergrund. Würden ihre Benutzer eine höhere Antwortzeit erfahren, dann würden sie sich vermutlich Gedanken um einen Wechsel zu einer anderen Plattform machen.

Die wirtschaftlichen Aspekte sprechen deshalb für einen PA-Ansatz.

Ausreichende Sicherheit bekomme man dann auch noch durch kluges Design (Ajax, Interaktionsstruktur, Navigationsstruktur, …) hin, so [Julian Browne](http://www.julianbrowne.com/article/viewer/brewers-cap-theorem).