

How Systems Complexity Reduces Uptime

By Ephraim Baron | May 05, 2021 | 0 Comments

Product Development Product Planning

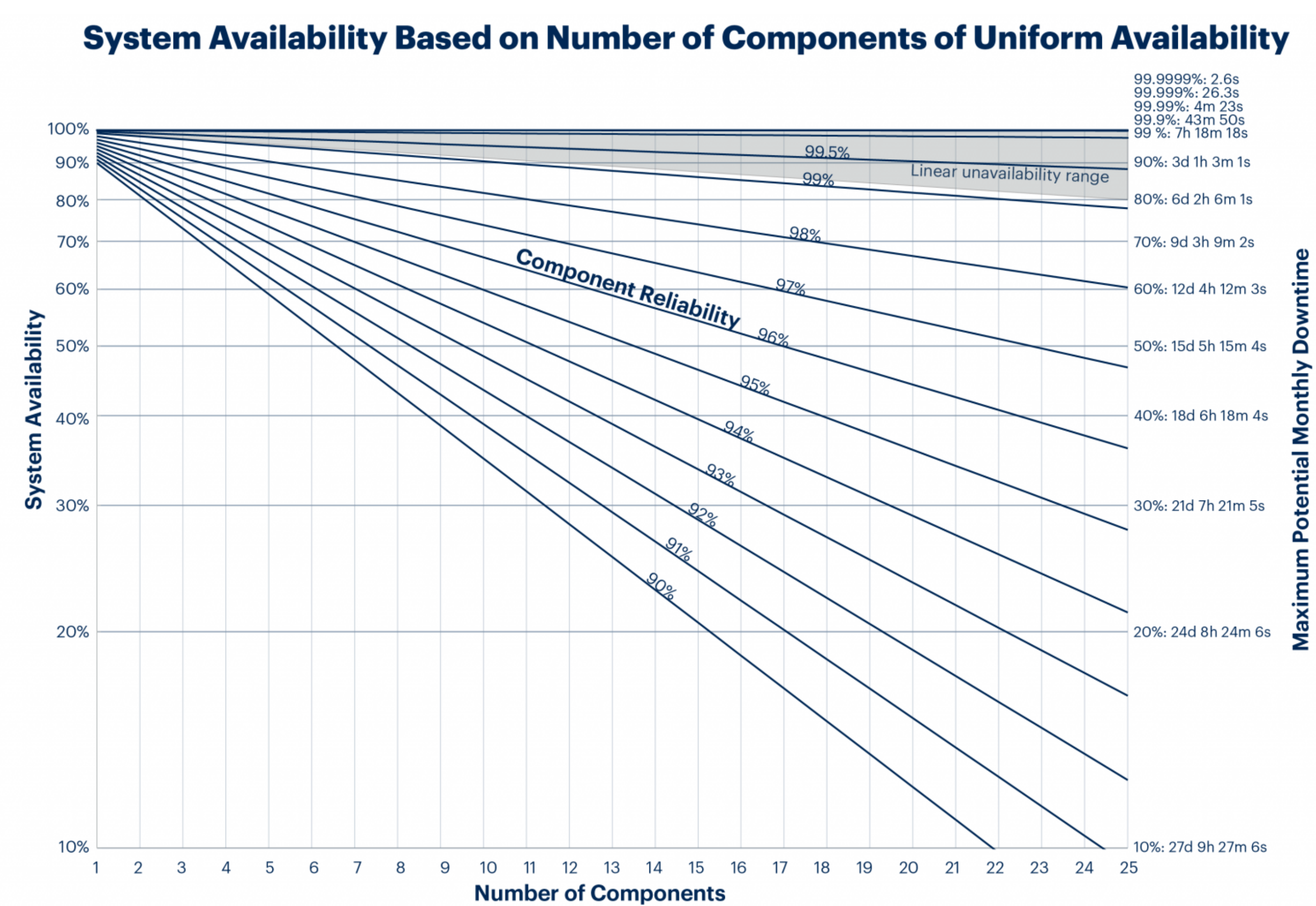
The pace of change in IT is unlike any other discipline. Even dog years are long by comparison. It’s an environment that makes us so primed for what comes next that we often don’t pause to ask why. This is very much the case with cloud computing, where cloud is the sine qua non of any CIO’s strategy and leads to vague approaches such as ‘Cloud First’.

This phenomenon also applies to application architecture, where microservices and composability are all the rage. The underlying rationale makes sense: break complex applications into function-specific components and assemble the pieces you need. After all, this is what software engineers do; they create and implement libraries of functionality that can be assembled in almost limitless ways. The result is an integrated collection of components are more elegant than the monolithic applications they replace.

Technical elegance, though, isn’t always better. To illustrate why, we need to turn to probability – not the complex stuff like Bayesian decision theory or even Poisson distributions. I’m talking about availability, which in IT means the percentage of time a system is able to perform its function. A system that is 99.9% (‘three-nines’) available can perform its function all but 0.1% of the time. Pretty good, right? Of course the answer to that question is, it depends. Some applications are fine with three-nines. For others, four-nines, five-nines, and even higher are more appropriate. How you would feel, for instance, about getting on an airplane if there were a 1-in-1000 chance of “downtime”?

From Component to Systems Availability

The aggregate availability of a system is the product of the availabilities of each component. For example, the availability of a system with 3 interdependent components, each having 99.9% availability, is 99.9% x 99.9% x 99.9% = 99.7%. The following figure illustrates the availability of a system with multiple components, each with identical availability. The number of system components is show on the x-axis, and the aggregate system availability is shown on the y-axis (plotted on a logarithmic scale). The maximum potential monthly downtime for a given availability level is included on the second y-axis.



For component availabilities greater than -99%, the following approximation is even simpler and goes as follows:

```

A_c = component availability
U_c = component unavailability
U_s = system unavailability
N_c = number of components
U_c = 1 - A_c [example: U_c = 1 - 99.9% = 0.1%]
U_s = 1 - N_c * U_c [example: 1 - 10 * 0.1% = 99.0%]
    
```

The availability of a system with ten components, each having three-nines availability, is reduced to two-nines, increasing potential system downtime from 44 minutes per month to 7 hours 18 minutes per month.

Implications for product managers, solutions architects, and CTOs

The point of this article is not to condemn composable applications; it’s to encourage thoughtful, intentional design. Just because everyone seems to be doing something doesn’t mean you must. In the case of application and service architecture, this means viewing the system holistically. In support, I offer the following:

- Occam’s razor:** entities should not be multiplied unnecessarily
- Albert Einstein:** Everything should be made as simple as possible, but no simpler
- KISS principle:** Keep It Simple ...

Key Takeaways

- Design for component failure and to minimize its impact.
- Make components decoupled and asynchronous whenever possible so that loss of one component does not cascade to others.
- Make critical components redundant and automatically scalable.
- Avoid stateful components whenever possible.
- Understand service interdependencies and failure modes.
- Eliminate unnecessary complexity (KISS).

Leave a Comment

Name

Email

Website

Save my name, email, and website in this browser for the next time I comment.

Comment

Post Comment

Search all blog posts

Direct to your Inbox

Sign up to receive our latest research highlights, expert blogs and news about upcoming events.

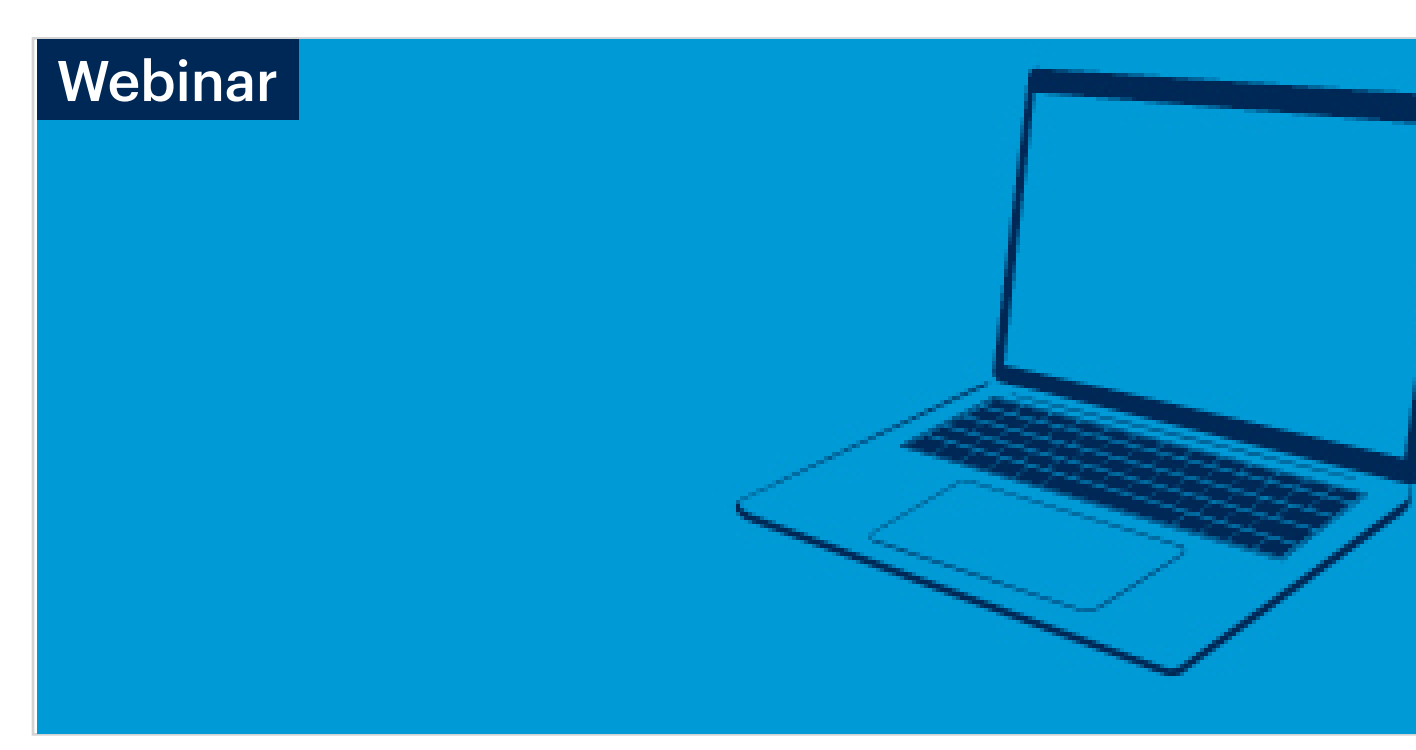
Business Email

Subscribe

By clicking the "Submit" button, you are agreeing to the [Gartner Terms of Use](#) and [Privacy Policy](#).

More from Ephraim Baron

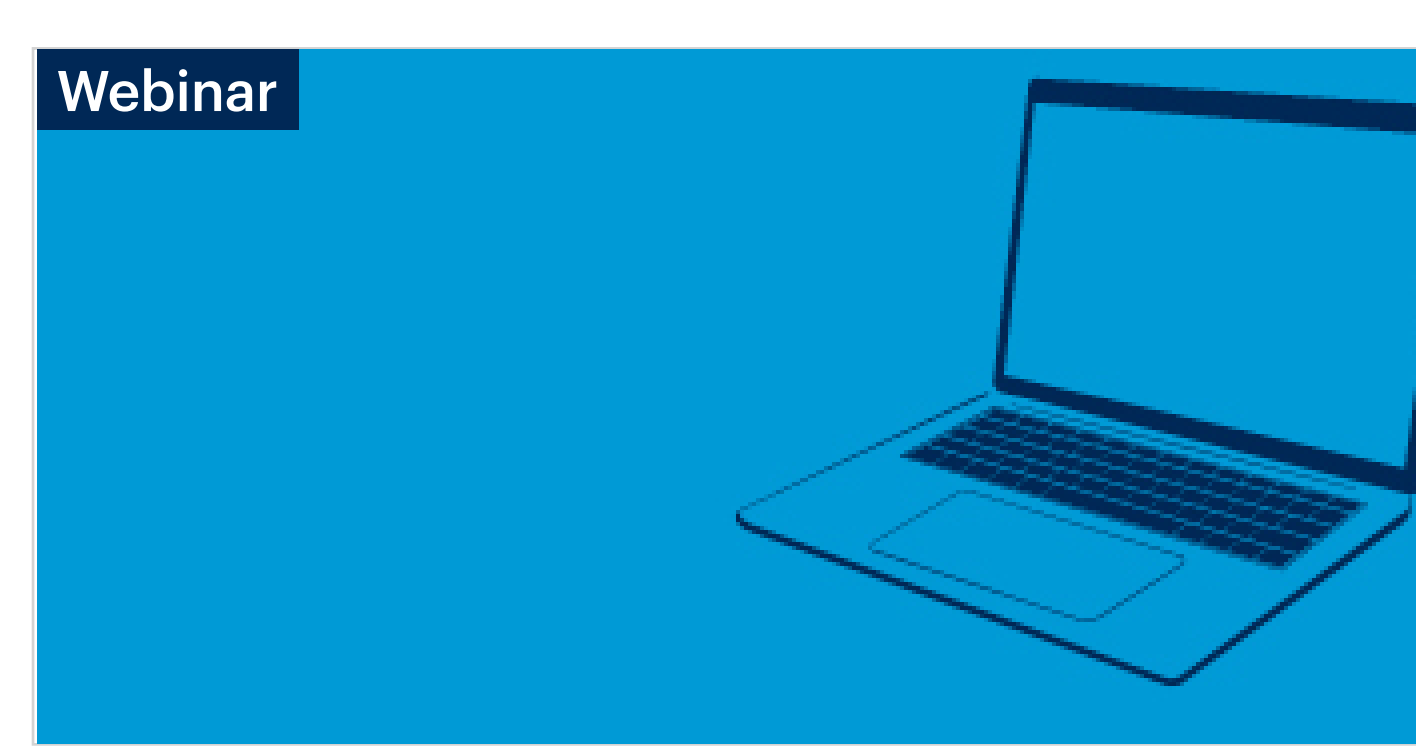
Dig Deeper



IT Spending Forecast, 1Q22 Update: The Russian Invasion of Ukraine

In this free webinar, Gartner reviews the latest changes to...

Watch Now



The Gartner Top Trends for Tech Providers in 2022

This free webinar identifies 10 trends that are of prime...

Watch Now

Solutions

- By Function
- Audit & Risk
- Customer Service & Support
- Finance
- Human Resources
- Information Technology
- Innovation & Strategy
- Legal & Compliance
- Marketing & Communications
- Product
- Sales
- Supply Chain

By Industry

- Education
- Energy & Utilities
- Financial Services
- Government & Public Sector
- Healthcare
- Manufacturing
- Retail
- High Tech & Telecom

Insights

- Featured Insights
- Benchmarking
- Cost Optimization
- Gartner Business Quarterly
- Strategic Planning
- Top Insights

Resource Centers

- COVID-19
- Diversity, Equity and Inclusion
- Reset Your Business Strategy Amid COVID-19

What We Do

- Research & Advisory
- Conferences
- Consulting
- Digital Markets
- Peer Insights