**Julio Pochet - Breaking Temporal Coupling**

I chose this topicbecause I’ve noticed it in real-world software and wanted to understand it better. Many programs force things to happen in a strict order when they don’t really need to, which can slow things down and make the code harder to maintain. This topic stood out to me because it explains how breaking that dependency can make software more efficient and flexible.

The main idea is that not all operations must happen in a set order—sometimes, they can run independently or in the background. The *Pragmatic Programmer* suggests using event-driven programming, message queues, or background tasks to keep things from being too dependent on timing. This can improve performance and make applications more scalable. A good example is how some websites let you upload a file and keep working while the file processes in the background instead of making you wait.

One place I’ve seen this in action is cloud services. Platforms like AWS and Google Cloud use **event-driven systems** where different tasks happen independently, and the system reacts as needed. Instead of running one thing at a time in a strict order, events trigger actions only when necessary. This makes things run smoother and prevents delays.

A helpful resource I found is **"What is an Event-Driven Architecture?"** By AWS. It breaks down how this works in cloud computing and how businesses use it to efficiently handle large amounts of data.

🔗 [**Read more here**](https://aws.amazon.com/event-driven-architecture/)