The **Byzantine Generals’ Problem** is a famous abstraction that serves to demonstrate one of the key problems in computer science, especially in regards to distributed computer systems (of which **cryptocurrency** is an example).

Imagine several Byzantine generals camped around an enemy city who communicate with each other only via sending messengers. They have to make a collective decision on whether to attack the city or retreat. However, some of the generals are traitors and may actively work against reaching a consensus. Is it possible to create a system that will ensure that the loyal generals decide on a common plan of action regardless of available knowledge on which generals are the traitors?

A thorough analysis of the problem shows that such an algorithm is possible, but only if more than two-thirds of generals are loyal.

Decentralized cryptocurrencies such as Bitcoin (BTC) are in essence distributed computer systems: their networks are composed of individual **nodes** operated by independent people or organizations that compete to process **transactions** and add them to the end of the **blockchain**. With nodes separated geographically and independent of each other or any central authority, it is impossible to reliably know which nodes are supplying faulty information about transactions — either maliciously or by accident.

Byzantine Fault Tolerance refers to the property of a distributed computer system that allows it to overcome this problem and consistently form **consensus** regardless of the fact that some nodes disagree with the rest, either accidentally or on purpose. This can be achieved via technical solutions, such as Bitcoin’s **proof-of-work** algorithm, but only if more than two-thirds of nodes remain loyal to the system.