

# Software Engineer 3.0

## paradigma baru *software engineering*

By Yanzero

# Cerita Siti dan Kripto

# Pengembangan *software* mengalami perubahan besar:

Dulu memprogram operasional bisnis:

1. Penjualan -> Point of Sales
2. Pasar -> Marketplace
3. Bank -> bank digital

Sekarang memprogram benda/asset:

1. Uang -> programmable money
2. Sertifikat tanah -> programmable RWA
3. Intellectual Property -> programmable IP

# Pengembangan *software* mengalami perubahan besar:

Dulu software menyelesaikan masalah inefisiensi terhadap operasional bisnis

1. PoS mempercepat kasir.
2. *Marketplace* mempercepat jual-beli.

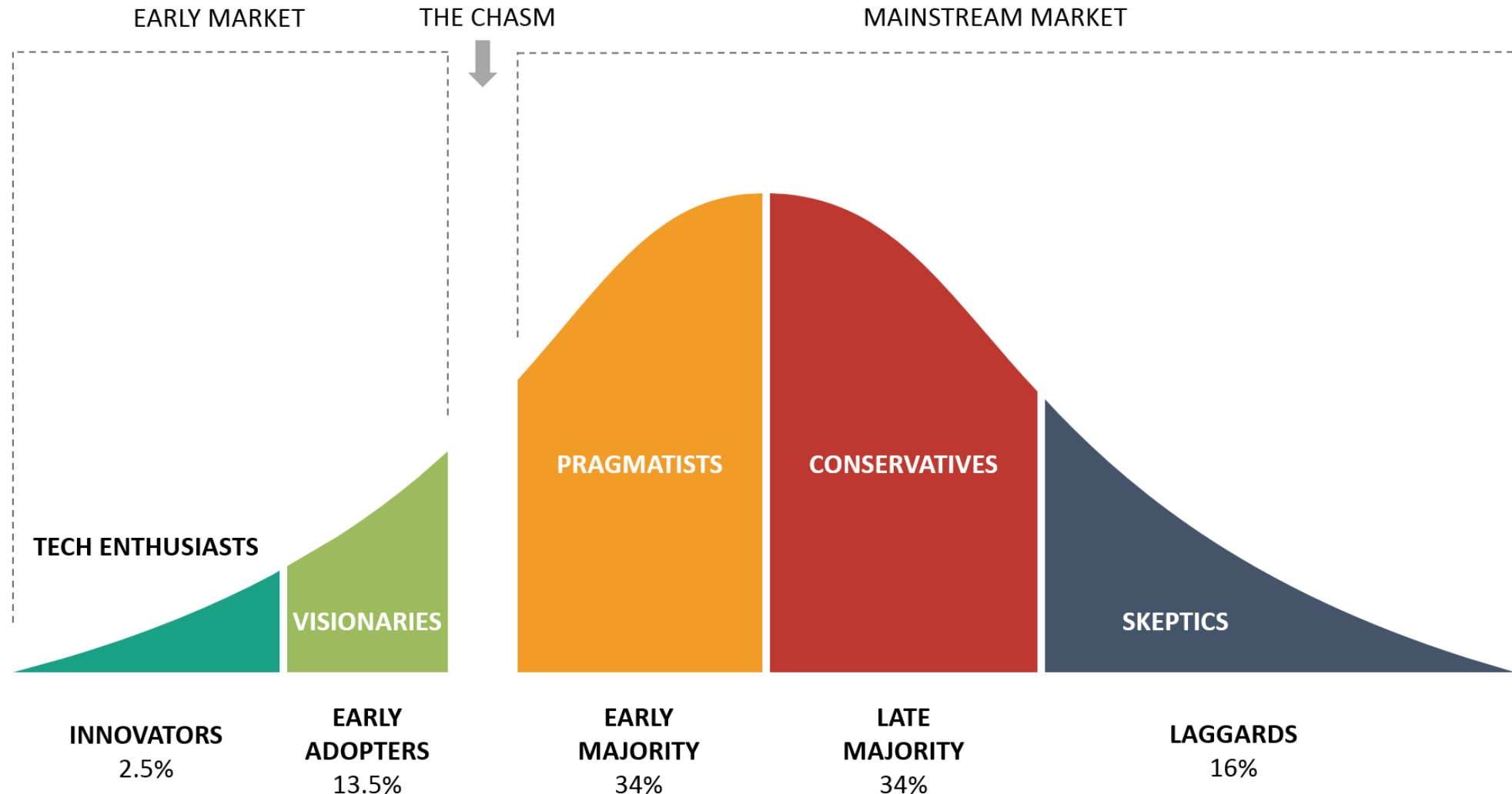
Sekarang software memiliki tantangan menyelesaikan masalah yang lebih besar seperti *human behaviour, fairness* dll

1. Mencegah korupsi sampai level kebijakan.
2. Mencegah *fraud* yang dilakukan orang dalam.
3. Menghilangkan kebijakan politis.
4. Membuat mekanisme harga yang fair bahkan aplikator tidak bisa mengatur.
5. Membuat organisasi yang demokratis bahkan foundernya tidak punya pengaruh.

Pahami tujuan besarnya.  
Software bisa membuat dunia ini lebih baik  
tanpa batasan teknis dan sosial.

Web3 meningkatkan derajat *software engineer* dari hanya menyelesaikan urusan teknis menjadi bisa menyelesaikan masalah sosial yang lebih besar dan lebih kompleks.

# Menjadi Early Adopter di Web3



# Teknologi baru itu

1. Di luar nalar masyarakat secara umum saat ini.
2. Banyak yang skeptis.
3. Belum sempurna sehingga masih ada ruang untuk berinovasi
4. Persaingan kerjanya sedikit.
5. Belum dikuasai AI.



Web3 masih baru dan belum sempurna  
sehingga ruang inovasi masih terbuka lebar.  
Reward sebagai inovator terlalu besar  
dibandingkan risk-nya.

# Jogja sebagai pusat inovasi teknologi Web3

1. Top 10 hackathon (internasional) AVS (Actively Validated Service): Veritrade, Jackramp dan CreatorHub
2. Finalist hackathon (internasional) zkTLS Reclaim : Marksense dan zkLink.
3. Finalist hackathon (internasional) Ethena - internet bond: ProtectFi dan Crediflex.
4. Disertasi doktoral tentang appchain berbasis Actively Validated Service dan Zero Knowledge Succinct Non-Interactive Argument of Knowledge (zkSNARK)

# Apa tujuan Web3 bagi dev?

Membuat segala sesuatu bisa diprogram tanpa batas secara *integrity*.

*Integrity* = sistem berjalan dengan benar tanpa perlu ada pengawasan.

Bitcoin = programmable money.

Smart contract = programmable contract.

NFT = programmable jpeg.

Story = programmable IP.

# Contoh tanpa batas

1. Program agar tidak bisa dikontrol oleh siapapun
2. Program agar orang dalam dan penciptanya tidak bisa fraud dan utak-atik
3. Program agar suplai terbatas
4. Program agar tidak ada intermediary
5. Program agar uang user tidak dititipkan kepada siapapun
6. Program agar immutable/tidak bisa diupgrade
7. Program agar siapapun yang menjalankan hasilnya benar
8. Program agar yang menjalankan server biasa siapapun tapi mereka gak bisa utak-atik

Apakah bisa di Web2?

# Apa saja yang bisa diprogram?

1. Money
2. Sertifikat tanah
3. Intellectual Property
4. Jpeg
5. Lagu
6. Organisasi
7. Apapun

# Use cases:

1. Sistem gaji langsung terkoneksi ke rekening perusahaan dan ditransfer perdetik.
2. Membuat uang perusahaan tidak bisa disalahgunakan.
3. Membuat organisasi yang keuangannya 100% dikendalikan anggota.
4. Membuat bank tanpa debt collector, tanpa menilai nasabah.

# Tujuan tersebut menghasilkan:

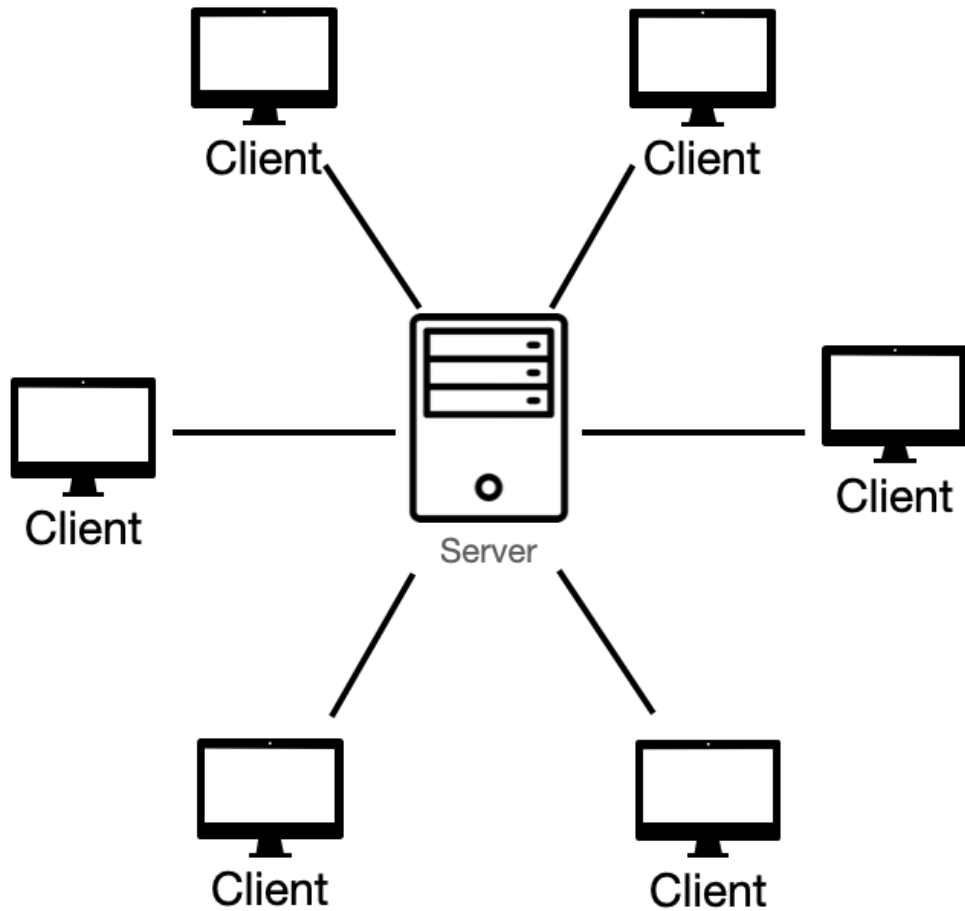
1. Decentralized -> P2P
2. Immutable -> konsensus
3. Trustless (bukan trusted) -> konsensus
4. Verifiable -> kriptografi

# 4 teknologi besar dalam Web3:

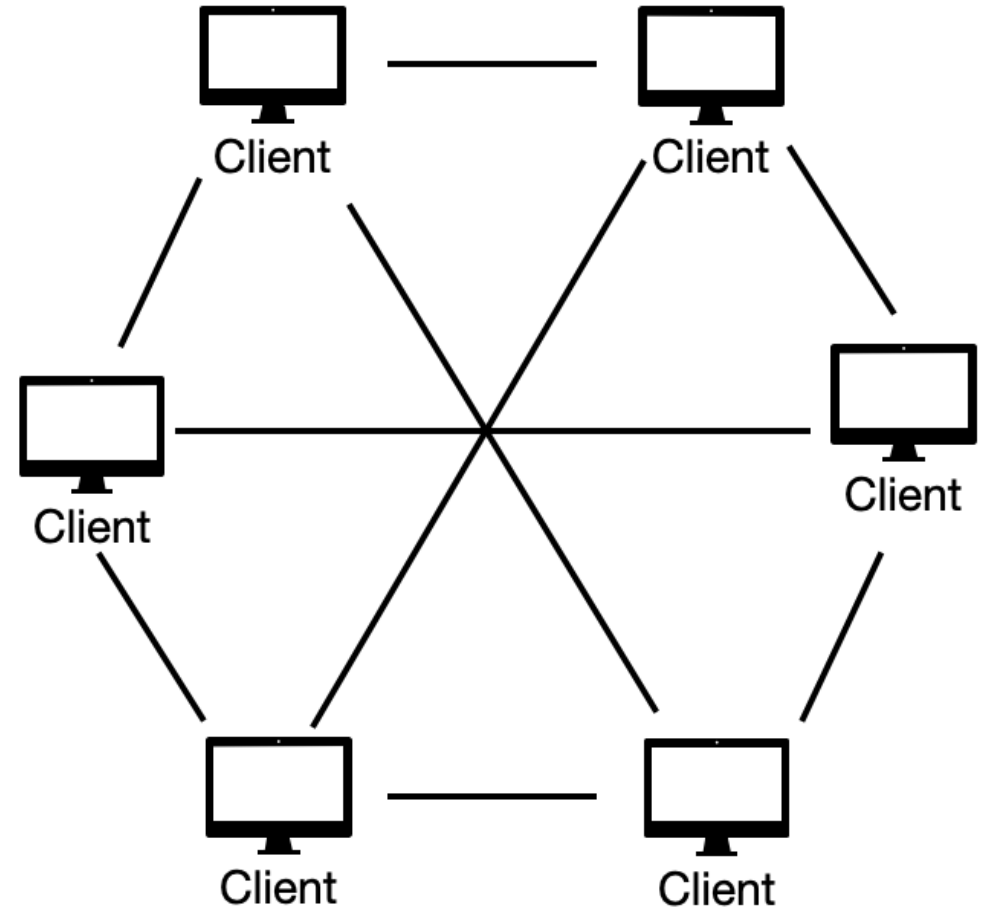
1. P2P networking
2. Konsensus
3. Kriptografi
4. Cryptoeconomics



# P2P vs Centralized



**Client Server Architecture**



**P2P Architecture**

# Contoh Decentralized

Server/node Bitcoin bisa dijalankan oleh siapapun mau itu pemerintah, badan intelijen, kepolisian, user secara langsung dll.

Saya sebagai user bisa menjalankan nodenya. Siapapun bisa menjalankan nodenya.

# Contoh Decentralized

Node tidak menyimpan password seperti di Web2.

Autentikasi menggunakan kriptografi *public dan private key symmetric*.

User memiliki kuasa penuh terhadap *user management*.

# Contoh Trustless

1. Sebagian server atau microservice Tokopedia dihosting oleh pemerintah
2. Sebagian server Bitcoin dihosting oleh pemerintah

Server Bitcoin dihosting oleh pemerintah = pemerintah tidak bisa mengontrol.

Server Tokopedia dihosting oleh pemerintah = pemerintah bisa kontrol.

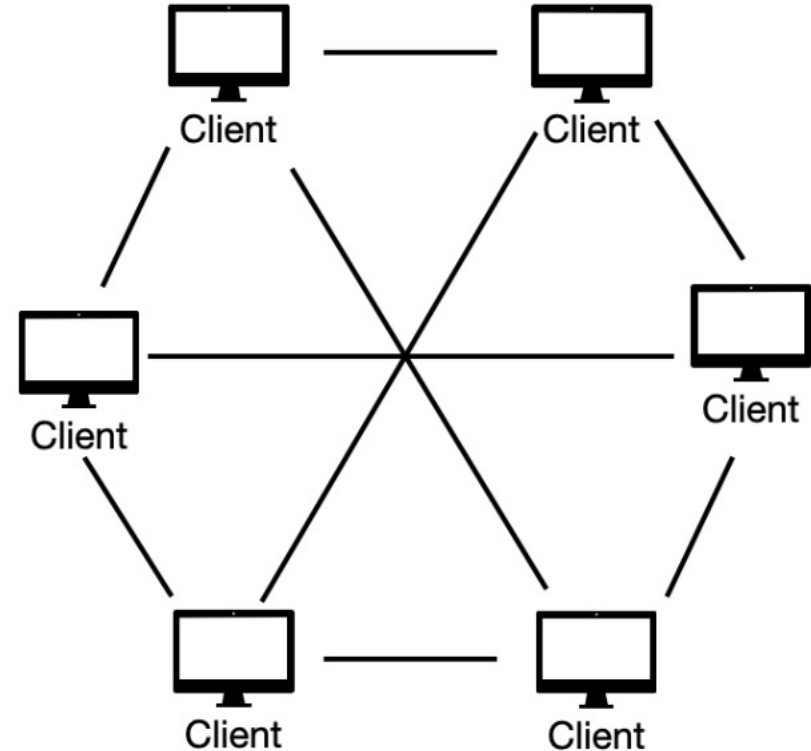
# Contoh Verifiable

- Setiap node bisa kita verifikasi dengan cepat menggunakan kriptografi.
- Setiap transaksi user bisa diverifikasi.
- Tahu node mana yang valid dan tidak.

# Contoh konsensus

1. Nakamoto consensus
2. Ethereum Proof of Stake : Gasper
3. Solana Proof of Stake : TowerBFT

Setiap pihak menghasilkan sebuah *agreement*.



# Konsensus

- Konsensus menentukan transaksi mana yang valid.
- Menentukan validator mana yang mendapatkan insentif dan disinsentif ( *cryptoeconomics* ) dan dieksekusi *realtime*.
  - Yang jujur dapat reward
  - Yang tidak jujur tidak dapat reward dan rugi (mengeluarkan biaya yang besar)
- Ethereum ada 1jt validator dengan network P2P mereka bisa mencapai sebuah *agreement* karena konsensus. Bayangkan ada 1jt komputer yang berkoordinasi secara P2P dan menghasilkan sebuah kesepakatan yang sama.

# Cryptoeconomics

Cryptoeconomics adalah mekanisme memberi insentif dan disinsentif/hukuman secara *monetary*/uang.

Cryptoeconomics memastikan bahwa tindakan korupsi biayanya lebih besar daripada keuntungannya..

Mekanisme insentif dan disinsentif ( *cryptoeconomic* )

- Yang jujur dapat reward
- Yang tidak jujur tidak dapat reward dan rugi (mengeluarkan biaya yang besar)



Jaringan P2P+ Konsensus + Kriptografi + Cryptoeconomics  
= menghasilkan *shared security*

# Mengenal *shared security*

*Shared security* artinya mekanisme di mana blockchain berbagi sumber daya keamanan bersama untuk memastikan semua pihak yang terlibat aman dari berbagai ancaman.

Menjamin :

1. Liveness : SLA uptime 99.999%
2. Anti-corruption : memastikan  $\text{cost fraud} > \text{profit}$

Banyak pihak

1. Banyak yang *stake*/menjaminkan
2. Banyak pihak yang terlibat

# Teknologi Web3 menghilangkan pengawasan

Web2:

Pemerintah diawasi DPR -> DPR diawasi rakyat -> rakyat diawasi siapa?

Web3

Tidak perlu ada pengawasan. Semua *trustless*.

Kembali lagi ke tujuan *integrity*.

# Seberapa efisien Web3 untuk dev?

Membuat aplikasi perbankan kurang dari 1 bulan yang dapat mengelola uang 1000 triliun bahkan lebih dan orang dalamnya pun tidak bisa korupsi (anti-fraud).

Bagaimana membuat sistem dengan mudah yang tidak bisa dikorupsi orang dalam? Hanya Web3 yang bisa.

Semua orang bisa menjalankan Node, ada banyak node di *network*, kita tidak harus menjalankan node.

Apa keuntungannya untuk dev?

# Bagaimana caranya dev smart contract hanya fokus ngoding?

- Tidak mikir user management -> decentralized
- Tidak mikir database -> decentralized
- Tidak mikir maintenance server -> decentralized, P2P
- Tidak mikir skalabilitas -> decentralized
- Tidak mikir pencatatan uang dan uang fisik -> decentralized
- Tidak mikir payment gateway -> decentralized
- Tidak mikir trusted atau tidak -> trustless

# Web3 adalah *serverless* sesungguhnya:

- tidak bayar server
- tidak perlu maintain database
- tidak maintain server
- tidak ngurusin skalabilitas server