

# Judg.io

SMART CONTRACTS POWERED BY SUBJECTIVITY

# Judg.io: Team

## Legal Expertise



**Jeremy Sklaroff**

- Penn Law and Wharton JD/MBA
- Davis Polk & Wardwell: Cryptocurrency and Law
- Author, *Smart Contracts and the Cost of Inflexibility*, Published in Penn Law Review
- Author of Coin-Operated Capitalism, under review in Penn Law Review

**Davis Polk**

UNIVERSITY of PENNSYLVANIA  
LAW REVIEW

## Biz Dev and Strategy



**Vivek Chauhan**

- Wharton MBA (Current); Bachelor and Master in Electrical Engineering from Indian Institute of Technology
- Business Development Intern at Viant, ConsenSys
- Deutsche Bank: Former Vice President, led Algorithmic Investment Products business.



NOMURA



## Engineering



**Aaron Diamond-Reivich**

- Wharton UG (Current): Dual Degree in Computer Science and Economics
- Protocol Engineer at Balance, ConsenSys
- Penn Blockchain Club: Creator and Teacher of course on Smart Contract Development



# Smart contracts need dispute resolution, but existing solutions fail

- Contracts go wrong. **A lot.**
- Drafting contracts is a balance between precision and risk. Subjectivity makes that easier.

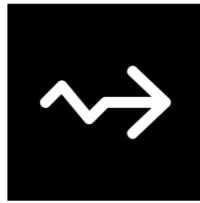
Exhaustive contract	Edge cases	Efficient subjectivity
<p>Visit at least 15 houses, smile, each conversation lasts at least 10 mins, shake at least 4 residents' hands, demo the product at least 6 times...</p> <p>Don't lie, manipulate the counterparty, willfully breach the agreement, interfere with counterparty performance...</p>	<ul style="list-style-type: none"><li>• What if there's no demand?</li><li>• What if Alice thought she was telling the truth about something at the time but was ultimately incorrect?</li></ul>	<p>Alice will use <b>commercially reasonable efforts</b> to sell the product</p> <p>Alice will perform the agreement in <b>good faith</b></p>

- Today's Ethereum smart contracts don't support subjectivity, raising the risk of disputes. "Master agreements" are the worst of both worlds.
- Existing dispute resolution is expensive. Companies pay \$300,000 per case for lawyers, judges, and administrative fees, creating a **\$3 billion dispute resolution market**.
- Current blockchain based arbitration models are closed platforms. No transparency. No accountability. No way to plan.

Current legal system and commercial contracts heavily relies on precedence which is ignored in current smart contract dispute resolution frameworks.

# Introducing Judg.io: Dispute resolution infrastructure for the blockchain economy

If we are aiming for a decentralized world with digitally native assets, we will need a **digitally native decentralized dispute resolution platform**



## Stable but decentralized

The platform converges on **key principles and decisions through precedent**, not centralized decision-makers. Judges select their own disputes. Fees are set by the market.



## Predictability through precedent

Parties understand the reason for the judge's decision and can **plan for similar disputes in the future**.



## Fast and affordable

**No more attorneys, no more courtrooms, and no more thousand-dollar fees** paid to rent-seeking arbitration bodies. Time limits chosen by the parties and coded into their smart contracts.



## Minimum viable disputes

Separate what's subjective from what's not. **Choose the scope of future disputes at the drafting stage**.

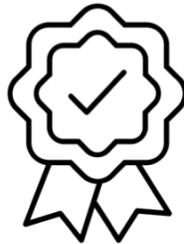


## Enforcement through blockchain

**Smart contracts are agreements and assets.** Existing dispute resolution requires parties to voluntarily join and arbitrate their claim. Decisions on Judg.io are automatically and immediately enforced.

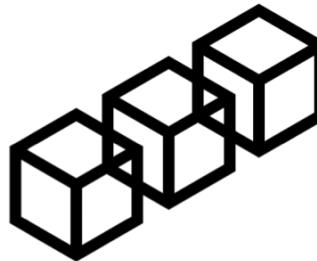
Judg.io is a **decentralized arbitration platform** for smart contracts that uses **precedent, reputation, and markets** to solve disputes quickly and cheaply and unleash commercial usage of blockchain

# Why Judg.io needs blockchain: Trust, precedent, and enforcement



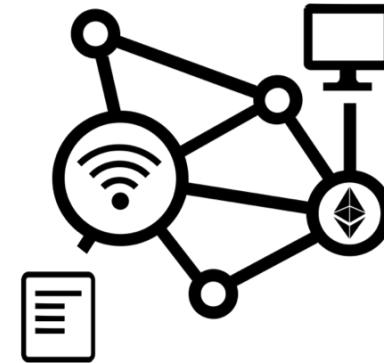
## Reputation and Trust

- Append-only database of rulings, shared between unknown and untrusted participants.
- Anonymity discourages malicious intent like bribery, collusion, etc.
- Judge reputation earned without interference by a central intermediary.



## Immutable Precedent

- Database of past rulings, allowing judges to link decisions to precedent.
- Immutability eliminates the ability for powerful parties to interfere in precedent creation.
- Precedent results in a set of key decisions and principles that contractors can use to plan their transactions. This transparency is not possible with current models.



## Smart Contracts and Enforcement

- Smart contracts combine agreement terms with an enforcement mechanism. This means that Judg.io can directly release contract payment or enforce refunds.
- Only litigation offers this kind of enforceability, but at an extremely high price. Cheaper solutions like online dispute resolution require parties to voluntarily join, arbitrate, and pay.

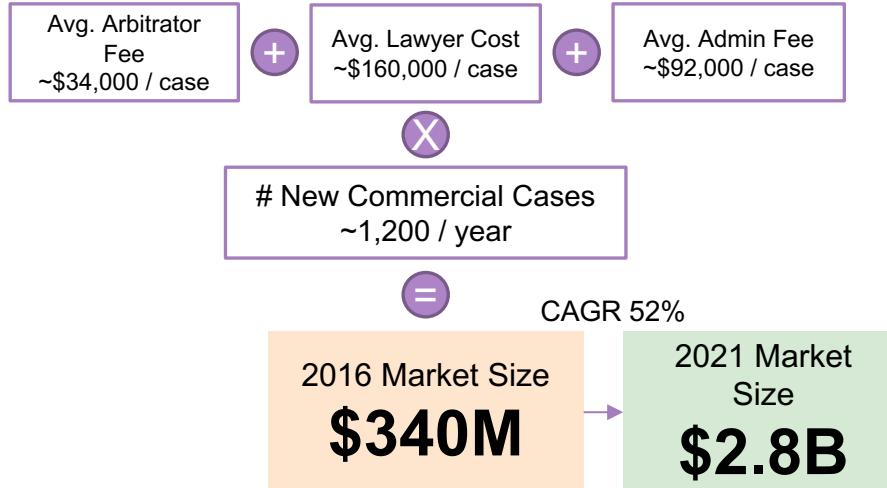
# Why Judg.io 1-> ICO governance: A distinct billion-dollar opportunity

- ~ \$5 billion in ICOs over past 18 months, but governance solutions are badly needed
- Dev teams want flexibility, investors want accountability
- “To receive remaining 50% stake of tokens, founding team will release **working code** before 1/1/19.”
- **Anonymity, decentralization, and judge-level incentives** mitigate conflicts of interest
- Fast on-chain arbitration tied to digitally native assets can aid much needed ICO governance while still preserving innovation and flexibility in development via subjective terms

**ConsenSys is leading ICO self-regulation efforts and possesses world-class legal, cryptoeconomics, and capital markets expertise**

# Why Judg.io 2 -> Arbitration market is massive, but it's not meeting customer needs

## The arbitration market is large and growing quickly



Above estimate doesn't even account international arbitration (\$3M per case) and new, blockchain-enabled markets (p2p commerce, supply chain etc.)

## Several customer needs are unmet by existing arbitration systems

Customer characteristics	Criteria
Small-medium sized enterprises	Annual revenues < \$100M
Low-medium legal sophistication	Total legal budget < \$15M
Using generic or boilerplate agreements	Scarce (or no) dedicated in-house counsel
Operating in industries affected by blockchain	Prevalence of manual, paper-intense processes

Judg.io's target industries: supply chain management, real estate tokenization, digital freelancing, online retail

# Validating assumptions: Judges

**Assumption:** Judges with no legal experience can accurately decide cases

## Results:

- ✓ Baseline accuracy > 50%, with 80% choosing the accurate justification
- ✓ Precedent boosted accuracy by 25%
- ✓ 33% of respondents referenced precedent to make their decision



## Expert interviews:

- ✓ Warren Hatch, President, Good Judgment Inc.
- ✓ Adam Sannit, Knowledge of Counsel, Norton Rose Fulbright

**Further boost in accuracy when filtered for education and experience**

**Conclusion:** Precedent improves legal reasoning, even for judges without legal education. That's a source of value not captured by competitor products.

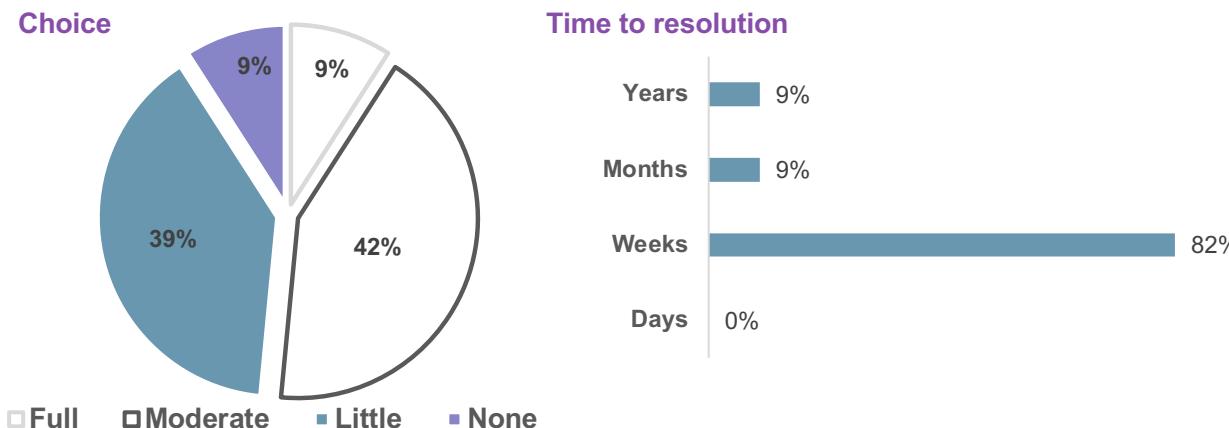


# Validating assumptions: Demand (customer)

**Assumption:** Existing dispute resolution isn't user-friendly – Judg.io solves a real need

## Results:

- ✓ Most-requested features: judge reputation (30%) and precedent (24%)
- ✓ 50% of respondents had little or no choice over how disputes were handled
- ✓ Most disputes weren't resolved for weeks, with some taking much longer



## Expert interviews:

- ✓ Chris Wray & Niranjan Sivakumar, Mattereum
- ✓ Jim Flannery, UGA Professor and entrepreneur working on blockchain dispute resolution
- ✓ Colin Rule, Founder, Modria.com (eBay's dispute resolution platform)
- ✓ Matt Peterman, Founder, InsurePal (p2p insurance)
- ✓ IBM dispute resolution

**Conclusion:** Judg.io uses the right set of features to improve a frustrating, time-consuming process and create competition in the dispute resolution market



# Customer acquisition: Litigants



## 1) Build and learn about customers

Partner with ConsenSys spokes in our target industries for customer development, early pilots, and testing.



## 2) Deploy across the ecosystem

Engage Ethereum dApp developers outside of ConsenSys to build Judg.io into products without dispute resolution, or to upgrade existing solutions.



## 3) Integrate with enterprise players

Work with forward-thinking companies experimenting with blockchain.



## ICO governance

Partner with ConsenSys spokes and industry education efforts.

Interest from consensys spokes and other ecosystem participants to partner with platform

# Customer acquisition: Judges



## 1) Standard Judges

- Recruit **students** from graduate programs in law, economics, computer science, psychology, etc., highlighting how easy it will be to **earn income as a Judg.io judge**.
- Engage the **Ethereum community** and encourage people to take part of a **radical experiment in blockchain governance**.

## 2) Expert judges

- Cultivate relationships with **legal academics, industry experts, and practicing attorneys**, encouraging them contribute their unique expertise in solving the highest-stakes disputes.



# Competitive landscape: Precedent makes us different

There are a range of blockchain dispute resolution products, but most are very early stage and restricted to specific types of disputes. None offer Judg.io's key features: published opinions, rewards for precedent, scalable solution.

	bisq	OpenBazaar	Delphi	Bitrated	KLEROS	JURY.ONLINE	Judg.io
API to integrate with other platforms				✓	✓	✓	✓
Uses a judge instead of jury	✓	✓		✓			✓
Reputation system for arbiters		✓	✓	✓		✓	✓
Standardized dispute resolution process <small>Time constraints, evidence procedures, etc.</small>	✓		✓		✓	✓	✓
Tokenized			✓		✓	✓	*
Judges get rewarded for sharing precedent							✓
Judges publish opinions							✓

\* We're still evaluating the need to tokenize our platform, and possible advantages/disadvantages.



# Design choices

## Preventing Bribery

### Time restrictions

- Judges are unable to claim cases until all evidence is submitted.
- Decisions must be entered within time specified by smart contract.

**Effect:** Minimize time that can be spent on arranging details of bribes. Incentivize judges to spend this time assessing evidence and preparing a useful rationale.

### Financial penalties

- Judges and users place a deposit during sign-up.
- Users can bring a bribery case in Judg.io, just like any other smart contract dispute.
- Any judge or user caught soliciting or offering a bribe forfeits their deposit.
- The party who successfully presses a bribery suit is rewarded with the briber's deposit.

### Social penalties

- Judges can only join the platform after being invited by an existing judge.
- Judges caught bribing lose their reputation. The judge who invited the briber also loses reputation.

**Effect:** Make collusion unstable. Any bribes could be "traps" to win a bribery suit against the recipient.



## Compensation

### Only pay for disputes

- If Alice and Bob decide not to file a dispute, then they pay nothing.



### Initial deposit and match

- When Bob files a dispute, he provides the default deposit amount: 2.5 JUDG tokens.\*
- If Alice fails to provide her deposit within the pre-agreed timeframe, she automatically loses. Bob receives his deposit and refund for the chairs.



### Paying for a highly-rated judge

- Suppose Alice wants a highly-rated judge. Alice can provide a larger deposit than the required minimum to attract expert judges.
- Deposit increases are only recoverable by the depositor. In other words, Bob cannot recover Alice's larger deposit, even if he wins.



### Winning and losing

- No appeal – Loser pays 1 JUDG to the judge and withdraws his remaining 1.5 JUDG. Winning party withdraws his full 2.5 JUDG.
- Appeal – Loser of appeal pays 1.5 JUDG. Parties can increase deposits to attract highly-rated judges.



## Judge or Jury?



### Judge

We assume that decisions made by a single judge are similar to decisions by a panel of judges. We're validating this assumption using Mechanical Turk.

**Network Effects**  
Stable precedent and greater predictability.

**Safety**  
No bribery or collusion.

**Cost & Reward**  
Balancing cost with rewards.

Jury-based system harnesses wisdom of crowds, minimizing variance in decisions due to individual interpretation.

Higher throughput. Cases can commence as soon as a single judge claims and decides the case, rather than waiting for multiple judges.

Raises coordination costs and bribery risks, multiplying the number of potential bribe monitors.

Parties face higher costs and possibly lower reward per judge.

\* Here, a token is being used to simplify the math. Deposits would work the same without a token. Instead of depositing 2.5 JUDG, the parties would deposit 2.5 times some baseline amount. For example, if the baseline amount were .0001 ETH, then initial deposits would be .00025 ETH, appeals would be .00015 ETH, etc.

# Design choices

## Reputation and selecting judges

Judge reputation is calculated using several factors



Number of decisions overruled on appeal



Assessments of rationale, quality of precedent by highest-rated judges



Proof of credentials (graduate school, Bar admission, employer, etc.)



Arbitrated many cases involving the same agreement type or industry



Reputation of other judges recruited to join the network

### Localized reputation

Each litigant selects his/her desired reputation function, deciding which of the above factors to weigh most heavily.

These functions are added together and the reputation for each judge is calculated.



- Gives users discretion over what qualities are important in a judge.
- Doesn't allow the judge to select specific cases where he/she has an interest in the outcome.
- Expensive to calculate new reputation for each case.
- Doesn't give judge flexibility in what types of cases he/she rules over.



### Marketplace approach

Judges with the highest reputation are given the first opportunity to claim cases. After a predetermined time, cases are opened to the regular judge pool.



- Allows the market to set fair value for litigation.
- Easy/cheap to calculate and store global reputation.
- Forces judges to rule on cases they might not have expertise or experience in.
- Market system gives judge maximum opportunity to seek out cases he/she has vested interest in.



### Direct assignment

The platform then assigns the judge with the top reputation to the case with the highest compensation.



- Judges do not have the ability to select cases that they have a vested interest in the outcome.
- Easy/cheap to calculate and store global reputation.
- Forces judges to rule on cases they might not have expertise or experience in.



# Success metrics

## Achieving Product Market Fit

- Reduction in time taken to resolve a dispute (at least 60% less than offline methods)
- Reduction in cost to resolve a dispute (at least 90% less than offline methods)
- Customer satisfaction in dispute resolution (NPS > 7.0)
- Average judge rating / reputation (> 9.0)
- % decisions upheld on appeal (> 90%)

## Achieving Scale

- # of cases resolved
- Average deposit amount per case
- User metrics, for both litigants and judges:
  - # of total active users on the platform
  - # of new users in the platform
  - % of new users acquired through user referrals
  - Acquisition cost for new users

# Product Roadmap



## Phase 1 (t + 3 months)

- Parties can embed platform into their smart contracts
- Ability to receive arbitration from anonymous, randomized judge or preselected judge
- Ability for parties to appeal rulings
- Standardized amount of evidence allowed
- Incentive structure to compensate judges (flat fee) and prevent bribery
- User interface for easy evidence submission, tracking of case, review of decision

## Phase 2 (t + 9 months)



- Ability for judges to link to precedent
- Upgraded reputation system for judges (SME-TCR), punishing bad behavior via economic penalty, increased reputation for cases frequently used as precedent
- Ability for litigants to pay extra for high quality judges
- Separate pool of expert judges and lay judges
- Integration with Oracle services for faster ruling

## Phase 3 (t + 15 months)



- Robust system for incentivizing judges to link to precedent, plus upgraded UI allowing judges to search for past cases
- Advanced compensation model, allowing arbitration to occur at real market value
- Create repository that publishes major new cases, so all users of platform are up-to-date with important rulings
- Promote smart contract standards per industry use case



# Financials and Assumptions

Monthly Financial Snapshot (every 6m)	Jun-18	Dec-18	Jun-19	Dec-19	Jun-20	Dec-20	Jun-21	Dec-21
<b>Users</b>	0	50	449	2,169	9,662	24,243	38,457	51,522
<b>Revenue (\$)</b>	-	-	44,928	433,718	3,864,873	9,697,182	15,382,932	20,608,991
<b>Cost of Sales (\$)</b>	-	-	31,450	303,602	2,705,411	6,788,028	10,768,053	14,426,294
<b>Gross Margin (\$)</b>	-	-	13,478	130,115	1,159,462	2,909,155	4,614,880	6,182,697
<b>Operating Expenses (\$)</b>								
Advertising & Promotion	-	5,000	6,993	26,686	41,649	46,649	46,649	46,649
Employee Benefits	-	-	7,725	9,525	12,410	15,963	19,823	23,814
Legal	5,000	5,000	2,000	2,000	2,000	2,000	2,000	2,000
Office Rent	3,000	3,000	4,320	6,221	8,958	12,899	18,575	26,748
Office Supplies	500	500	1,000	1,000	2,000	2,500	2,500	2,500
Salaries	28,667	28,667	51,500	63,500	82,733	106,417	132,150	158,762
Other	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000
<b>Total</b>	42,167	47,167	78,538	113,932	154,750	191,428	226,697	265,473
<b>Operating Income (\$)</b>	(42,167)	(47,167)	(65,059)	16,184	1,004,711	2,717,727	4,388,183	5,917,224
Income Before Taxes (\$)	(42,167)	(47,167)	(65,059)	16,184	1,004,711	2,717,727	4,388,183	5,917,224
Income Taxes (\$)	-	-	-	5,664	351,649	951,204	1,535,864	2,071,028
<b>Net Income (\$)</b>	(42,167)	(47,167)	(65,059)	10,519	653,062	1,766,522	2,852,319	3,846,196

## Revenue

Positive unit economics in **18-24 months**.

User growth modelled based on growth of market participants and S-curve.

Judges **keep 70% of platform fee**. Aim to increase even higher.

Fee starts at \$0 and with time **grows to \$400** (<< traditional arbitration).

## Costs

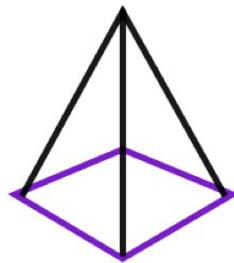
Leverage team's tech and management experience to reduce expenses.

Hire 2 full time devs to work on front/backend, paid at market rate.

Access to mentors for legal, game theory, and business development.

Marketing and advertising expense is included.





Judg.io

SMART CONTRACTS POWERED BY SUBJECTIVITY

# How it works?



# How Judg.io works: n-sided market for human judgment

**Network effects drive value for all users.** Judges compete to boost their reputations by providing useful, influential judgments. Parties use past cases to write better contracts, and can fall back on dispute resolution to solve edge cases.

**Greater certainty.** Alice knows that damage to at least 35% of chairs **will** be a breach of merchantability. She also knows that less damage, particularly wobble and finish defects, **may** be a breach.

**Greater precision.** If Alice believes she can reliably keep her defect rates down, she can simply write a smart contract that specifies, “no more than 35% of the shipment will be affected by wobble.”

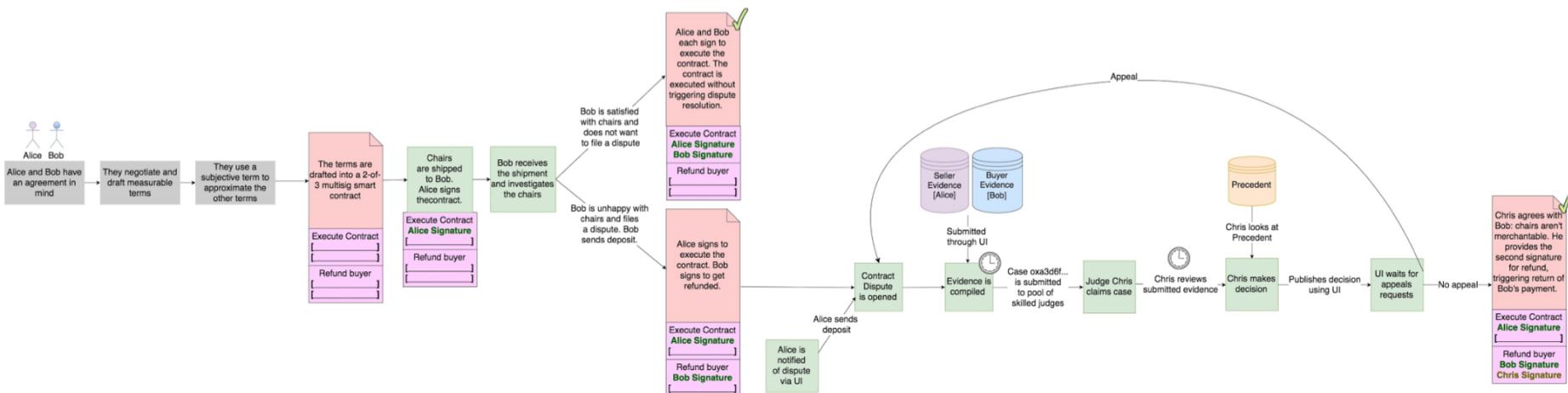
**Improves the entire ecosystem of smart contract writers.** The judge’s decision is published and can be reviewed by other users as they draft their own smart contracts:

- Sellers of **chair manufacturers** know that 35% is a threshold for merchantability.
- Other **furniture manufacturers** know that 35% is very likely to be relevant.
- Manufacturers of **any good** know that the 35% threshold might be relevant.

**Decentralized consensus.** Over time, the platform stabilizes around key decisions and principles, even though each judge acts individually. Bonuses for linking to past cases are a cheap but effective way to generate precedent.

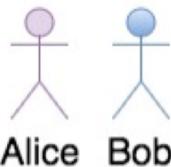
# How it works: Alice and Bob

Alice manufactures chairs. Bob is a retailer. Bob wants to buy 500 blue chairs from Alice.



# How Judg.io works: Creating the smart contract

**Alice and Bob have their agreement in mind. So they draft a multisig smart contract to capture all of the measurable terms of the agreement.**



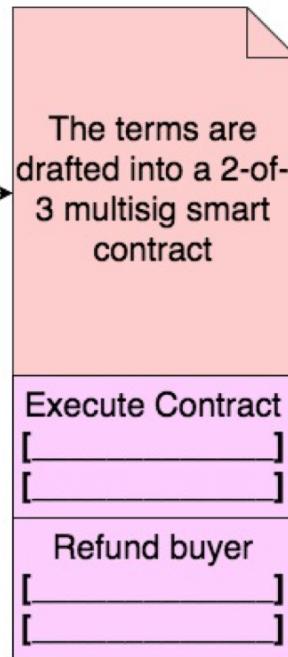
Alice and Bob have an agreement in mind

They negotiate and draft measurable terms

Contract term	Source of data
The chairs cost 0.1 ETH	Payment of ETH verified by wallet
Alice must meet certain production milestones	Alice's manufacturing facility validates deadlines (eg, wood has been cut, chairs have been formed, chairs have been painted, etc.)
The chairs must be blue	Alice's manufacturing facility validates color (blue paint was used)
Alice must ship the chairs by a certain date	UPS verifies pickup
The shipment must weigh the right amount	Smart-scales in Alice's factory validates weight
The shipment must not be subject to sudden drops or exposed to moisture	IOT devices and/or UPS measure in-box conditions

# How Judg.io works: Creating the smart contract

They use a subjective term to approximate the other terms



**Bob wants to be confident that the chairs will be good quality. But he can't define that parameter in a precise way.**

- Instead, Bob specifies the chairs will be of **merchantable quality**.

**To use Judg.io, some additional terms will be needed:**

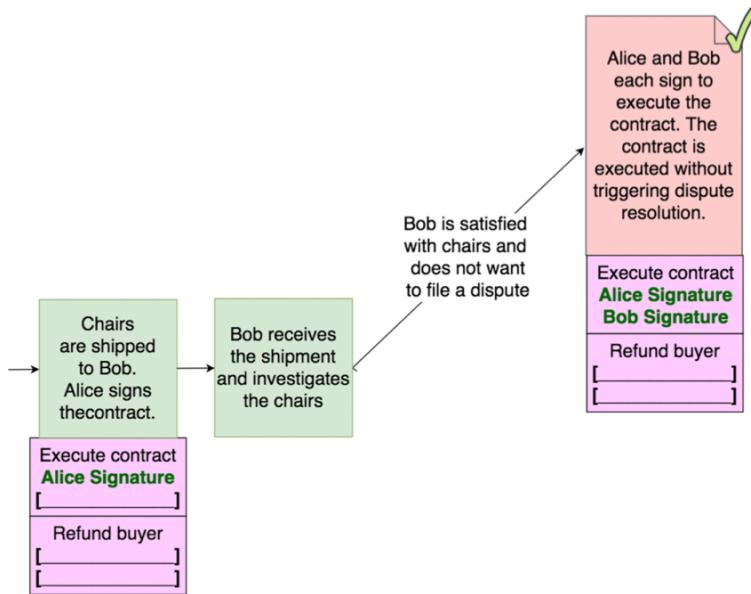
- How much time the parties have to dispute the contract.
- Code that conditions full execution of the smart contract upon receipt of 2 signatures.

Contract term	Source of data
The chairs must be of merchantable quality	Evidence collected by Alice and Bob, and shared with Judg.io
Bob has 5 days after receipt to dispute the shipment	UPS confirms Bob's receipt; oracle timer measures days after receipt
The parties have 2 days after the initial judgment to file an appeal	Judg.io timer launched automatically after initial judgment entered

**Every Judg.io contract has at least two possible outcomes. Each requires two signatures to execute.**

- Scenario 1:** Both Alice and Bob agree either that the terms of the contract were met or they agree that the terms of the contract were broken. They either both sign "execute contract" or "refund buyer."
- Scenario 2:** Alice and Bob disagree on the fulfillment of the contract. The third signature comes from Judg.io.

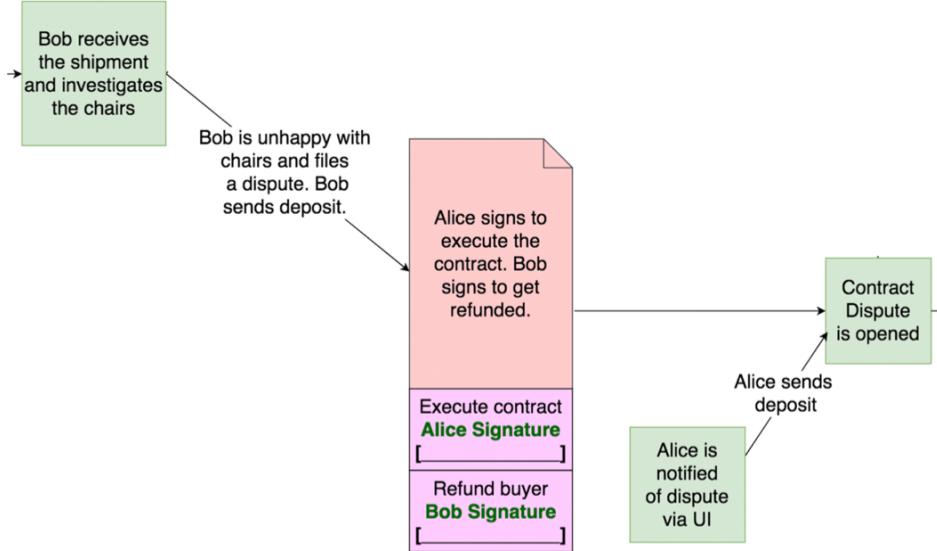
# How Judg.io works: Accepting the transaction



**Most agreements will proceed without dispute.**

- Alice signs the smart contract when she ships the chairs.
- Bob examines the chairs (within 5-day time allotment) and agrees they're merchantable. He also signs the smart contract.
- Smart contract payment term triggered.

# How Judg.io works: Filing a dispute



## But what if Bob disputes the shipment?

- Bob thinks the chairs are wobbly and many have nicks in the color finish.

## Bob opens a dispute and provides a deposit to the system.

- Bob has an opportunity to provide a higher deposit than the default amount. This will incentivize higher quality judges to pick the case.

## Alice is notified of the dispute through a UI. She provides a deposit as well.

- Alice also has the opportunity to increase the deposit.

## What if Alice fails to provide a deposit?

- At the end of pre-agreed timeframe, she loses the dispute.

# How Judg.io works: Filing a dispute

Through the UI, the parties are asked to provide:

## 1) Anonymized Narratives

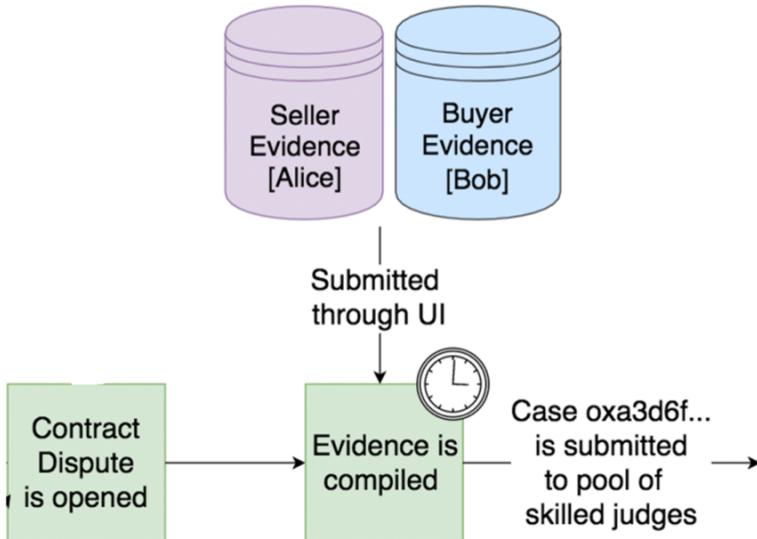
“Seller” [Alice]: *I’ve had over 1,000 successful chair shipments since starting my business. Very low return rates. I did not deviate from my normal production process.*

“Buyer” [Bob]: *I looked at 20 chairs. 7 were wobbly and 3 had nicks in the finish. I can’t sell these to customers, therefore they’re not merchantable.*

## 2) Other relevant evidence

- Data from Alice’s construction and painting machines
- List of past transactions showing data about when buyers signed off

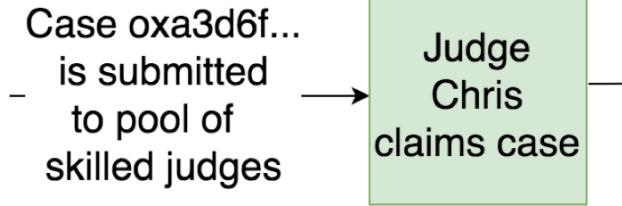
- Video of Bob testing the wobbly chairs
- Photos of nicks in the paint



The data is stored using IPFS.

The dispute is now added to the list of open disputes.

# How Judg.io works: Judge dynamics



**Judge Chris browses the list of open disputes and chooses Alice and Bob's dispute.** Because Alice and Bob only provided default deposit, they may receive a lower-rated judge or experience a longer wait before their dispute is reviewed.

**Most judges will look for highest-fee disputes.** But they can also look for interesting disputes, browsing narratives submitted by the dispute filers.

**Access to disputes is based off judge reputation.** Higher rated judges get earlier access to newly filed disputes. Two important incentives at play:

- Judges want to raise their reputation, so they have earlier access to high-fee cases.
- Filers want to provide good fees so that their dispute is seen and selected by highly-rated judges.

**We're exploring a number of ways to generate judge ratings**

- Other judges rate the rationales provided by the deciding judges.
- "Would you recommend this judge to other users?" Yes/No.
- Judge metrics: length of rationales, rapidness of decisions, frequency of using precedent.
- Bonuses for judging a highly-cited case.

# How Judg.io works: Making a decision

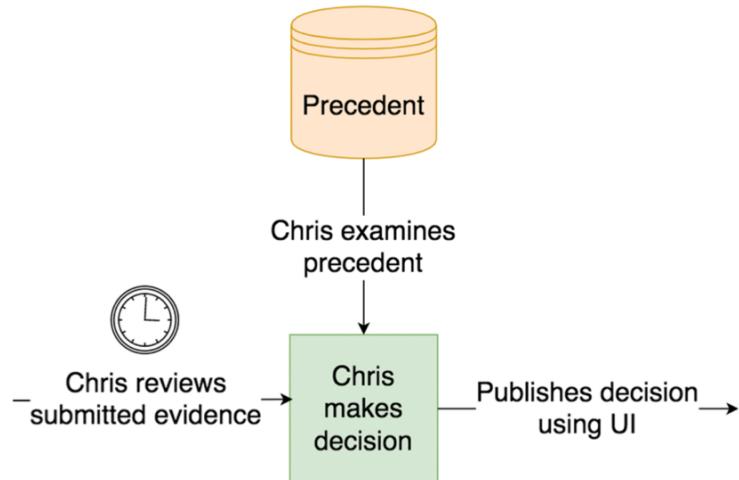
**Chris assesses Alice and Bob's evidence and narratives**

**Chris makes a decision** within the timeframe specified in the Judg.io smart contract.

- Judge decides that there is enough damage to the chairs that they're not merchantable.

**Chris publishes the rationale and stores it on IPFS.** This creates transparency and preserves privacy as narratives are anonymized

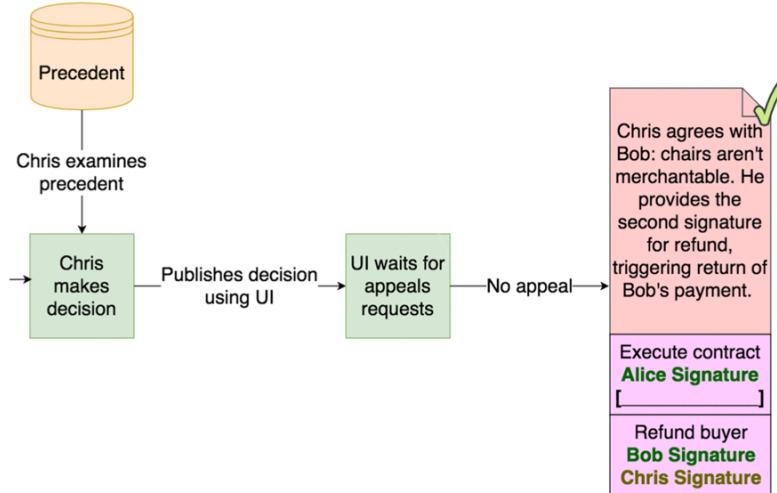
Chris: *I confirmed the wobble and paint flaws via evidence. The wobble affected 7 of 20 examined chairs, and the nicks affected 3 of 20 examined chairs, implying that at least 35% of the chairs were defective. As a result, I rule that the chairs were not "merchantable."*



**Chris receives bonus for linking his decision to a previous case**

Chris: *I referenced case A91048BzJ. There, a buyer examined 16 tables out of a shipment of 200, and found that 8 were wobbly and 6 had damaged finish, implying that at least 50% of the tables were flawed. The judge in that case ruled that the shipment was not merchantable.*

# How Judg.io works: Finalizing judgment



**Since Chris has decided the chairs weren't merchantable, he signs the smart contract using the “refund buyer” function.**

**Alice and Bob have 2 days to appeal the decision.**  
If neither does, then the 2 day window elapses and Chris's decision is final.

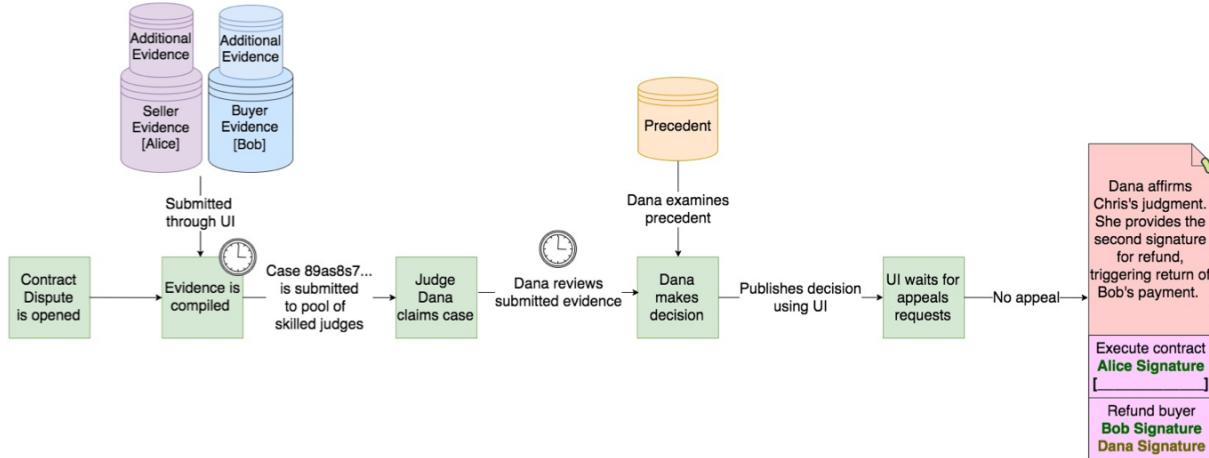
The funds are now unlocked for Bob to withdraw.

# How Judg.io works: Requesting an appeal

**Alice or Bob can request an appeal through the UI.** The appeals process follows the same flow as the initial judgment. The appeals judge, Dana, can review evidence Chris collected, or request additional evidence. Dana also publishes a rationale and receives a bonus for using precedent.

**To proceed with the appeal, the appeal requester provides an additional deposit.** Only judges with sufficiently high reputation are eligible to judge appeals. Judges are incentivized to grow their reputation so they can judge higher-paying appeals cases.

**Dana agrees with Chris that the chairs aren't merchantable.** Dana's decision is final. Once she enters her decision, the smart contract terminates.



# Risks

- **Growth in smart contract ecosystem:** Our platform is heavily dependent on growth in the overall smart contract ecosystem (particularly in the priority areas we have identified) and the adoption of smart contracts to replace traditional contracts.
- **Preference for on-chain dispute resolution:** The value of our platform rests on our hypothesis that many customer segments will prefer blockchain-based dispute resolution to traditional arbitration or litigation.
- **Incentivizing user participation:** Given we are creating an n-sided decentralized marketplace, we need to achieve simultaneous growth on both disputer and judge supply side. This may initially involve intense offline effort and high user acquisition costs.
- **Aligning incentives to prevent manipulation:** We have given a lot of thought to the game-theoretic aspects of the platform design to prevent issues like bribery, collusion, and self-interested judgments. However, we will need to closely and continuously monitor these aspects to ensure the integrity of judgments.
- **Technical challenges:** It will be critical to ensure data confidentiality and platform scalability, while maintaining transparency and integrity on the platform.