

# How can we make our data work better for us?

Ark 51 – turning documents to data with AI



Exporting that data into the CDM format

# The problem

- Financial entities will enter into a whole array of legal contracts with counterparties for trading.
- These contracts on their own are complicated enough.
- But with enough contracts, the information contained within begins to have an interdependent life of its own.

# The problem

- What happens if a country's credit rating from a specific agency goes from AAA to A+?
- What if my Net Asset Value drops 15%, how many contracts am I in breach of?
- Are any of my contracts under Russian Jurisdiction? What happens now?
- Can't answer the question without the requisite data.

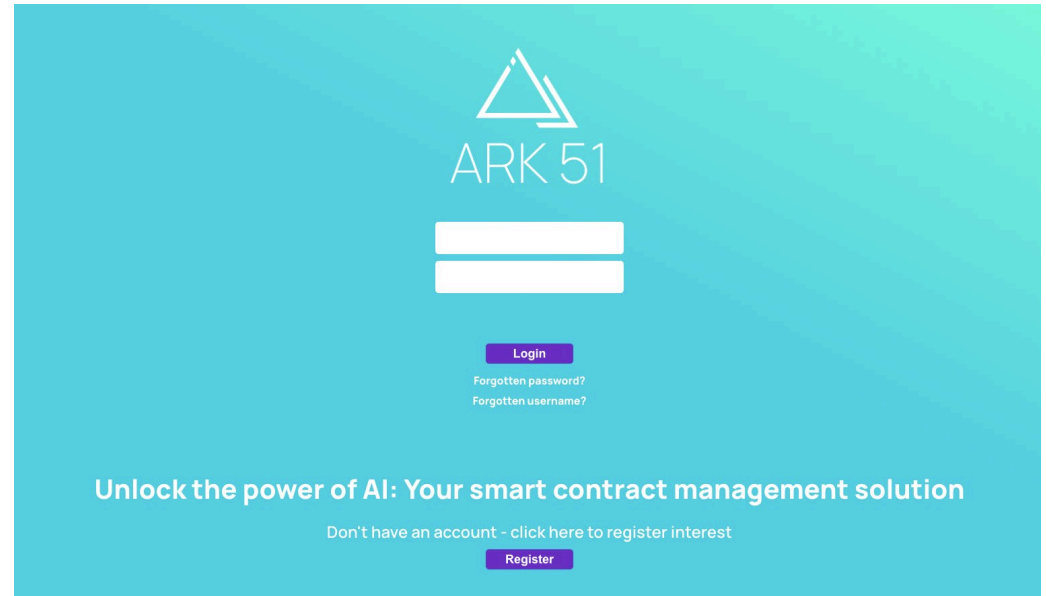
# The problem - (the real one!)

- Everyone knows that the data inside documents is massively valuable in risk tolerance assessment, future planning, collateral management.
- Extracting the data contained within the contracts into a digital format to ask those questions takes **time** and **money**.
- In most cases **too much time** or **too much money** to make it a justifiable business venture.



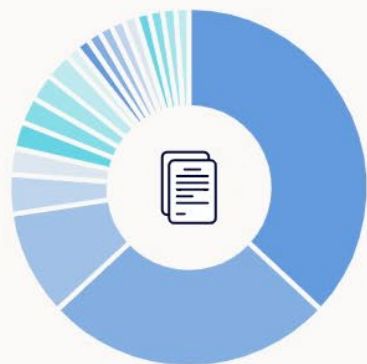
# How can Ark 51 help?

- (This is not a sales pitch!)
- Ark 51 has been designed to take a contract, and run it through a bespoke AI pipeline, built by cutting edge software developers working with in house legal experts, to take a contract, and quickly digitise the important questions contained within.





Upload



# Where does the CDM fit in?

- The CDM is the way to get systems to easily talk to one another.
- Once we all speak the same language, we inspire greater collaboration (and competition!)

```
"trustSchemeAddendum": false,
"baseAndEligibleCurrency": {
  "baseCurrency": "EUR",
  "baseCurrencyTerminationCurrency": false,
  "eligibleCurrency": [
    "EUR",
    "GBP"
  ],
  "eligibleCurrencyInclBaseCurrency": true
},
"creditSupportObligations": {
  "minimumTransferAmount": {
    "partyElection": [
      {
        "amount": {
          "unit": {
            "currency": [
              "EUR"
            ]
          },
          "value": 50000
        },
        "party": "Party1"
      },
      {
        "amount": {
          "unit": {
            "currency": [
              "EUR"
            ]
          },
          "value": 50000
        },
        "party": "Party2"
      }
    ]
  }
}
```

# Technical (don't fall asleep!)

- Our mapping is written in typescript in a node service hosted on AWS
- Strong typing of the CSA Elections went nicely hand in hand with the strong typing of the typescript language.
- Native mapper, built from scratch

```
957
958 export const csaToCdmMapper = (ark51Data: SystemDatasetIdentifierWithValue[], isVMCSA = false, isIMCSA = false, contractualPa
959 // First we need to extract some data for parties identifiers
960 const [partyAWithMeta, partyBWithMeta] = contractualParty;
961 const partyAIdentifierInfo = constructCDMCSAPartyFromContractualParty(partyAWithMeta);
962 const partyBIdentifierInfo = constructCDMCSAPartyFromContractualParty(partyBWithMeta);
963
964 // Generic function to extract data based on systemId
965 const findSystemIdValue = (id: string) => ark51Data.find(({ systemId }) => systemId === id)?.value;
966
967 // Generic function to extract multiple data based on an array of systemIds
968 const findAllInTable = (idArray: string[]) => ark51Data.filter(({ systemId }) => (idArray.includes(systemId)));
969
970 // Retrieve the credit support offsets (this must be present or handled)
971 const creditSupportOffsetsFromDataset = findSystemIdValue('creditSupportOffsets');
972 let creditSupportOffsets = false;
973 if (creditSupportOffsetsFromDataset) {
974   if ((creditSupportOffsetsFromDataset as string[][0]) === 'Applicable') {
975     creditSupportOffsets = true;
976   }
977 }
978
979 // Retrieve the holding and using posted collateral (this must be present, but only applies to NY CSA types)
980 let holdingAndUsingPostedCollateral: CDMHoldingAndUsingPostedCollateral = {
981   partyElection: [
982     {
983       party: CDMCounterpartyType.Party1,
984       useOfPostedCollateral: false,
985       eligibilityToHoldCollateral: {
986         eligibleCountry: [],
987         partyTerms: CDMHoldingPostedCollateralEnum.AcceptableCustodian
988       }
989     },
990     {
991       party: CDMCounterpartyType.Party1,
992       useOfPostedCollateral: false,
993       eligibilityToHoldCollateral: {
994         eligibleCountry: [],
995       }
996     }
997   ]
998 }
```



# What next?

- 2025 plans include adding legacy CSAs once finalised
- We've already helped and contributed to the CDM, we plan on continuing to do this and to continue advocating for as strong a typing per agreement type as possible
- Already have the AI capacity for GMRA, GMSLA and ISDA master agreement as well as a plethora of other contracts, ready to map to CDM as effortlessly as the CSA!



# Paul Hands

Chief Technology Officer

Happy to answer any questions or queries, please get in touch!

**E:** paul.hands@parallel-51.com

**M:** +44 (0)7935 729 540

