SHTAR — שטר

Symbolic **H**ierarchical **T**erms **A**nd **R**ules

A financial product definition language

SHTAR is a financial product definition language. Its central idea is a product as a **chronological system of records and payments**: observations establish facts, contractual conditions determine obligations, and deliveries settle those obligations. Earlier records determine which rights and payments remain.

A definition establishes what the product means, including the state it retains and the rules for updating that state. Processing a trade, valuing it under a model and implementing those operations in software are interpretations of that definition.

Contents

1 Purpose: preserve the chronology of a product 2

2 Define an illustrative autocall 2

3 Payables and the choice of valuation units 3

4 Define the state needed for the next step 4

5 Define the information a product needs 5

6 Interpret a definition for processing or valuation 5

7 Market status and model valuation 7

8 Define decisions independently of their valuation 9

9 From a product idea to a definition 9

10 Current application 9

A About SHTAR 10

SHTAR

1 Purpose: preserve the chronology of a product

A financial product can pay at several dates, remember an earlier observation, acquire a new obligation or finish with claims still awaiting settlement. Its definition must preserve these relationships through time. A single number at maturity does not, by itself, describe them.

Consider a coupon paid in June and principal due in December. By September, the coupon is part of the product's settled history. It cannot be delivered again as a December payment. The recipient might have retained the asset, reinvested it or spent it. Each choice leads to a different later holding; the original contract does not select that strategy.

An unpaid claim is different. If the June coupon became due but was never settled, that obligation may still be outstanding in September. Its date alone does not make it a completed payment.

SHTAR keeps these distinctions in the product definition: what was observed, what became payable, what was delivered, and what remains. Accounting can apply the definition to actual records; a pricing model can evaluate outstanding and contingent obligations using the information currently available.

A valuation can express dated payments as contributions at a common valuation date. That calculation is separate from a strategy that turns earlier receipts into future wealth. Combining payments into a single terminal amount requires the relevant holding, reinvestment or financing rules to be supplied.

Modularity supports this chronological description. Separately defined securities and sources of information can be combined while retaining their own rights, records and payment terms.

2 Define an illustrative autocall

Consider a note with principal 100 and a coupon of 2 per review. Its coupon is payable when the observed index is at least 70% of its initial level. It automatically redeems at an eligible review when the index is at least 100%.

The SHTAR definition retains one coupon-state variable, M: **the aggregate amount of missed coupons**, initially zero. It retains neither a list of those coupons nor past index observations. Principal, coupon amounts and barriers are terms; the review event supplies its timestamp and the context supplies the current fixing.

At each review before maturity, the definition specifies the coupon payable and the continuing state separately:

- **Coupon condition fails:** emit no coupon payable; the new state is $M + 2$.
- **Coupon condition succeeds:** emit a coupon payable of $M + 2$; the new state is zero.

Starting from zero, two missed reviews leave states 2 and then 4, with no coupon payment. A subsequent review at 85% emits a coupon payable of 6 and leaves state zero. The amount 6 is an output payment obligation, not part of the continuing coupon state.

If that successful review instead reaches 101% at an eligible call date, principal is also payable and the note has no successor. At maturity, the terminal rules determine principal repayment and any remaining coupon entitlement.

Each continuing review also returns the next observation request. For this note, it identifies the next review timestamp from the contractual terms. A terminating review returns no next observation.

The payable records its asset, quantity, recipient, obligor and settlement time. Settlement may occur several days after the review. **Coupon memory resets when the payable is created**, so a later review cannot include that amount again. The separate claim remains outstanding until settled; resetting M does not erase it. Settled payments remain in the historical record.

3 Payables and the choice of valuation units

SHTAR treats currencies, shares, bonds and other deliverable assets uniformly. A delivery identifies the asset, its quantity and its contractual terms. Currency can be a convenient reporting asset, without having a special place in the product language.

The contract specifies what is owed. A promise to deliver one share on a stated date remains that promise whether its value is expressed in currency, bond units or another supported asset. Choosing a reporting asset does not replace the share with cash or change its delivery date.

The concept of a **payable** therefore separates the contract's delivery terms from the numeraire chosen for valuation. No particular currency or other asset has a privileged role in the language. Exchange values and valuation conventions establish how a payable is valued in the chosen units; a mathematical numeraire and pricing measure belong to the valuation interpretation.

A payable can be **direct** or **synthetic**, depending on the market assumptions accepted in its valuation context:

- A **direct payable** has an exchange value treated as market-agreed and sufficiently liquid. Representing or projecting that value in a model may still require substantial computation.
- A **synthetic payable** has no such accepted market-agreement basis in that context. Its value is determined under a selected model, without assuming that market participants agree on it.

Different models can assign different exchange values to the same synthetic payable. These are valuations of the same contractual rights and obligations.

A bond or swap defined from underlying payments can be treated as direct. Both direct and synthetic payables may require analytic calculations, PDE or Monte Carlo. Their status depends on the market context and can change over time; it is not fixed by the security's type, its construction or the complexity of its valuation.

SHTAR

3.1 Securities can themselves be payable

A product can deliver a separately defined security. An option, bond, swap or dated claim can therefore be a component of another product's obligations. Its own definition explains the rights and obligations that the recipient acquires, including settlement, obligor and any later exercise or delivery terms.

3.2 Continuing a product and delivering a product

A swaption can become a swap after exercise. A compound option can become a vanilla option. The resulting product can have different terms, state and future obligations.

Delivering a security is a different relationship. The recipient acquires a separate claim, while the parent's own remaining obligations are determined by its definition. Transferring an existing security preserves the identity of the right being transferred.

These distinctions matter for both accounting and valuation. Ending a parent's lifecycle leaves its previously delivered claims outstanding. Reading information about another security does not acquire that security's payments.

4 Define the state needed for the next step

Chronology does not require repeatedly reading the entire past. A SHTAR product definition explicitly specifies its **state**, its initial values and its update rules. Two definitions can express the same economic contract while choosing different ways to retain the relevant history.

4.1 Minimal coupon state

The autocall example declares only the aggregate M as its coupon memory. A review needs that amount and its current fixing to determine the next coupon payable and updated memory. Payment records and the observation history are separate from this continuing state.

This is an optimization in the SHTAR definition itself. A different definition could retain individual missed coupons, but the example chooses the aggregate explicitly; the interpreter follows that choice. The definition remains independent of the model supplying projected observations and valuing its payments.

One aggregate suffices here because all remembered coupons have the same future treatment. Different expiry, accrual or eligibility rules may require additional declared state. The example keeps only what its own rules need.

4.2 An Asian average: update a running sum

Suppose an Asian option uses the arithmetic average of specified index fixings. A SHTAR definition can explicitly declare a running sum and count, both initially zero, and the following update for each new contractual fixing:

$$Q_{k+1} = Q_k + S_{t_{k+1}}, \qquad n_{k+1} = n_k + 1, \qquad A_{k+1} = \frac{Q_{k+1}}{n_{k+1}}.$$

SHTAR

Here Q is the running sum, n the count and A the average. Fixings of 100, 110 and 90 give a sum of 300 and a count of 3. A new fixing of 120 changes them to 420 and 4, giving an average of 105. The product need not recover the earlier fixing values to recompute that average.

Another SHTAR definition could explicitly retain every fixing. The sum and count are a sufficient summary for this averaging rule, chosen when defining the product rather than inferred by its pricer. Other contract terms may require additional declared state, such as a running maximum or whether a barrier was crossed.

4.3 Discretization and the historical record

For an average over specified fixing dates, this recurrence evaluates the agreed discrete average exactly in mathematical terms. A continuously averaged contract requires an explicit numerical integration approximation. SHTAR still specifies the retained state and its update rules; a numerical interpretation supplies the supported approximation without substituting a different product-state definition.

At a valuation date, accepted past fixings establish the starting state. Each projected continuation starts with that same historical sum and count, then updates them using its own future observations. Past observations are not simulated again.

Compact processing state does not erase the underlying record. Fixings and settlement records can remain available for audit or reconciliation without being carried in every pricing path. Each accepted fixing contributes once, including when a record is replayed or several products read the same average.

5 Define the information a product needs

A product may depend on a fixing, a calendar, a running average or the observable state of another product. SHTAR describes these dependencies through **contexts**.

A context supplies information in answer to typed questions, called keys. The key specifies the kind of answer expected. A closing-level key can return an index level; a missed-coupon key can return an integer; an exchange-value key can return units of one asset per unit of another.

This separates the product's need for information from the source that supplies it. The same definition can use accepted historical observations in one interpretation and projected observations in a valuation interpretation.

5.1 Contexts can have their own definitions and state

A context is a product that provides information and makes no payments. A running-average context, for example, can declare a count, a sum and their update rules, and supply the resulting average.

A security can combine that synthetic state with market information. An average-based note might use the running average alongside the latest index close and its own initial fixing.

A paying product can also expose a read-only view of its state. Another product may use that view without acquiring the first product's obligations or processing it a second time.

5.2 Several answers, nested views and derived information

Some questions have one answer; others have several. A request for one contractual fixing needs a single resolved value. A request for a portfolio's component exposures can return several entries for explicit aggregation. Equal exposures from two components remain two entries.

Contexts can be nested. A portfolio context might supply an asset-specific context, which in turn supplies a close or exchange value.

Contexts can also derive information from other contexts. A normalized index level can be calculated from an observed close and an initial fixing. A shared average can be combined with several market observations.

The source and observation time remain meaningful through these combinations. A derived view cannot reveal tomorrow's fixing as information available today.

6 Interpret a definition for processing or valuation

A definition can support several interpretations while retaining its economic meaning.

An **operational interpretation** applies the contractual rules to accepted observations, decisions and settlement records. It determines obligations that have arisen, deliveries that have settled, the product rights that remain, and the next relevant processing event.

A **valuation interpretation** determines the value of those rights and obligations from the information currently available. Unknown future observations and model-dependent values require a selected model and valuation method.

An **implementation** realizes an interpretation in a particular system. It can choose physical storage for the declared state, such as tensor packing, while preserving that state's meaning and update rules. Choosing a coupon list or an aggregate belongs to the SHTAR product definition, not to the interpreter or pricer.

The model retains a separate freedom: it can recognize a payable and use a suitable valuation formula or solver. Preserving the product definition does not require evaluating every future payment through

simulation.

6.1 An event-based operational interpretation

One useful processing interpretation represents the current trade by its terms and state. At a relevant event it determines:

1. The deliveries created by the contractual rules.
2. The continuing product and its updated state, if any.
3. The next observation or event requiring attention, specified by its time or a request for the next eligible occurrence.

For the autocall, a missed coupon changes the remembered amount. A successful coupon observation creates a dated claim. Automatic redemption creates the remaining claims and ends further note reviews.

Every continuing review returns a next-observation request. It may identify a timestamp from an observation table, determine that time from the current state, request the next eligible observation, or await another external event. The next request follows from the definition and current information; a universal engine schedule is unnecessary.

The continuing state can change shape when the product changes. A swap following exercise of a swaption needs the swap's own state. It need not retain the option's old fields.

6.2 Timestamps and the next available observation

Time in SHTAR identifies an **instant**, with a timestamp rather than only a calendar date. A date in the terms needs its observation or settlement convention, time of day and time zone to identify the intended instant. Several observations can occur during the same day. Review time, receipt of a record and settlement time remain distinct.

For a continuously monitored product, a definition can return an **epsilon** request: observe again as soon as the next eligible observation is available. In a live setting this may mean the next market tick. Epsilon expresses the finest available observation frequency; it is not a specified number of seconds or a floating-point increment. Each review returns a fresh request while the product continues.

A valuation model can interpret this observation rule through continuous monitoring, without reproducing every operational tick. Its solver then chooses an appropriate numerical approximation. Treating monitoring as continuous does not establish that every possible tick, hourly or other discretization gives the same price; the approximation must be suitable for the product.

6.3 Model time is a separate coordinate

A model will commonly convert SHTAR timestamps to floating-point time coordinates. The origin, units and model-clock mapping belong to the provider or model, not to the product definition. A request for the next observation therefore does not prescribe a numerical integration step.

Contractual calendar and accrual conventions still belong to the terms. For example, the day-count rule used to calculate a coupon is distinct from the coordinate used to integrate a stochastic model.

6.4 Consistent interpretation of shared components

Suppose two products depend on the same running average. Both should see the average for the accepted fixing. Two readers must not cause that fixing to be counted twice.

The same principle applies to obligations. Reprocessing an already accepted event must not create a second copy of the same payment.

A product's state also remains distinct from the market model's state. A contractual change from option to swap does not replace the model's volatility factors or market history.

7 Market status and model valuation

Direct or synthetic status describes the accepted basis for a payable's exchange value. The method used to compute or project that value is a separate choice. A direct payable can be computationally demanding, and a synthetic payable can have a simple valuation formula.

7.1 Direct payables

A direct payable's exchange values are accepted as market-agreed and sufficiently liquid within the valuation context. That agreement is an explicit assumption of the context. Values may be supplied as observables, derived from other liquid payables, or represented and projected through a model under those conventions.

Suppose, under matching delivery conventions, one share exchanges for 50 units of USD and one bond for 100. The bond's exchange value is then two shares. A separate bond-to-share quote is unnecessary because the value follows from exchanges already known.

USD is simply the intermediate asset in this example. Shares, bonds or another supported direct asset can also provide a reporting basis. A derived context can expose these calculations for reuse.

The values must refer to the actual payable. A share delivered today and a promise to deliver that share later can have different exchange values and obligations.

A bond in a short-rate model may require a nontrivial calculation of its value. Swaps, VIX futures and options, and other liquidly traded options can likewise require analytic calculations, PDE or Monte Carlo while being treated as direct. Accepted market values and the model's calculated values remain distinguishable; a difference between them can be a model-fitting error rather than a change in the payable's market status.

7.2 Synthetic payables

A synthetic payable lacks an accepted market-agreed exchange value in the bound context. Its value is determined under the selected model without assuming market agreement. Different models or valuation assumptions can give different values for the same contractual rights and obligations.

An analytic formula, Monte Carlo or PDE can provide a model valuation. An analytic answer remains model-dependent; computing it quickly does not make it an observed exchange value or an agreed market price.

A model-produced value can become an input to another product through a context. Its assumptions and source remain relevant. Computing, caching or reusing a number does not itself establish market agreement.

7.3 The same product can change status

A bond can be treated as direct while its market value is accepted as sufficiently liquid and agreed. Later, when it is off the run, that acceptance basis may no longer hold. The same bond can then be treated as synthetic (indirect). A previously issued option can undergo the same change when its market becomes less liquid.

Age alone does not determine the change: the relevant market conditions and assumptions do. The

product keeps its identity, terms, state and payment obligations. Its valuation context reassesses the status and the supported way to value it. A restored basis for market agreement can also support direct treatment again.

7.4 swaption delivers a swap: the model chooses how to value it

Suppose a SHTAR swaption specifies delivery of a separately defined swap when exercise is accepted. The payable is that swap, with its remaining rights, obligations and declared state. The swaption need not specify a formula or numerical method for valuing it.

One model may recognize the swap and have a closed-form expression for its value in terms of the current model state, the swap's terms and its declared state. Where these inputs contain the necessary information, the model can evaluate the payable immediately without projecting each remaining swap payment.

Another model may need to solve a PDE, branch into conditional Monte Carlo continuations from the current information, or use another supported method. These are alternative ways to value the same payable.

This is optimization within the valuation interpretation. The model can exploit a simpler expression for the swap's value while preserving the swap's definition and actual deliveries. These computational choices are available whether the swap is treated as direct or synthetic. Returning a value does not settle the swap in cash, and the swap's value and its expanded payment values must not both be counted.

SHTAR therefore provides two distinct freedoms: a product definition can explicitly choose an efficient state representation, and a model can choose a suitable way to value its payables. Neither requires the product definition to depend on a particular valuation model.

7.5 Connect a definition to the available capabilities

Before applying an interpretation, its provider establishes the market-agreement and liquidity assumptions, required inputs, and supported computation methods. This binding step is called **registration**. It can bind the delivered swap to a supported model formula or numerical solver without changing what the swaption delivers. Those bindings must remain suitable as the context changes.

Missing observations, settlement conventions or model capabilities remain explicit requirements. The product definition does not change merely because a particular provider cannot yet interpret all of it.

Each obligation contributes once to a valuation. Its supplied or computed value and its expanded future receipts must not both be counted.

A direct asset's observed value today also leaves its future exchange values unknown. A model may need to project those future observations when valuing a contingent product.

7.6 Numerical results and hedges

The remaining economic claims eventually yield numerical values in a chosen reporting asset. A future delivery of 100 units is not automatically worth 100 units today; the exchange and settlement conventions establish the conversion. A settled past receipt is part of history, not an additional payment by the remaining contract.

A Monte Carlo interpretation can aggregate numerical contributions across its realizations. Another supported method can evaluate the same definition differently.

Regression controls based on forwards or options can improve precision. When a control corresponds to a tradable payoff with a consistently known price, its fitted coefficient can be expressed in hedge units. A synthetic-state predictor can help estimation without representing a tradable hedge.

8 Define decisions independently of their valuation

Exercise, conversion and other party choices belong to the product definition. It identifies the deciding party, permitted alternatives, timing and obligations associated with each choice.

Operational processing can apply an accepted decision to determine the resulting rights and obligations. Until the choice is accepted, the alternatives remain pending.

Valuation needs an additional treatment of that choice, such as an exercise policy or solver. Defining an exercise right does not by itself determine how a holder will use it or what that right is worth.

The same distinction applies when a decision delivers a new security or changes the continuing product. The contractual alternatives remain reusable across supported accounting and valuation interpretations.

9 From a product idea to a definition

A useful starting point is the product's chronology: which observations and decisions matter, when obligations arise, how they settle, and what rights remain after each step.

Explicitly define the state, its initialization and its updates. A coupon definition may choose individual entries or an aggregate; an averaging definition may choose a fixing list or a sum and count. Check that the chosen state retains every distinction affecting future obligations. These choices provide optimization within SHTAR while leaving the valuation model independent.

Separately defined components can then organize the product. A reference-level context, coupon rule, dated claim or option can retain its own definition and be used by several products.

The resulting definition can be reviewed against the source terms before a particular processing or valuation interpretation is selected. Equality at a barrier, coupon memory, settlement timing and final redemption deserve explicit treatment.

AI can assist with this translation and reconciliation, including proposing a more efficient SHTAR definition. The selected state and update rules must be explicit before interpretation or compilation. The proposed definition should keep a reviewable connection to the source terms and identify unresolved clauses or inputs.

The product can then be bound to a provider that supports its required observations, market assumptions and valuation methods for direct and synthetic payables. Definitions remain available for other interpretations as those capabilities develop.

10 Current application

The current prototype demonstrates operational processing and saved-calibration valuation for SPX memory autocalls and selected European options using a Markovian rough-volatility model.

SHTAR supports Monte Carlo valuation for selected products with exercise rights.

These implementations cover subsets of the broader language. General-purpose compilation, unrestricted component dependencies and valuation of arbitrary discretionary product combinations remain further implementation work. Supported dates, payables and valuation assumptions must be bound explicitly. General intraday and epsilon requests are not yet implemented. The Asian example illustrates state representation; it does not claim an implemented Asian pricer.

The language reference gives the precise definitions and proposed syntax. The SPX pricing application describes the current concrete valuation profile.

A About SHTAR

SHTAR. Draft 0.1, 25 October 2026.

SHTAR