

A Logic for Resource-sensitive Coalition Games

Pinaki Chakraborty^{1*}, Tristan Caulfield^{1*}, and David Pym^{2,1*}

¹ University College London, England, UK pinaki.chakraborty.22@ucl.ac.uk,
t.caulfield@ucl.ac.uk, david.pym@sas.ac.uk

² Institute of Philosophy, University of London
England, UK david.pym@sas.ac.uk

* corresponding author

Abstract. Reasoning about strategic ability in multi-agent systems typically relies on strategic logics such as Coalition Logic and Alternating-time Temporal Logics, which model what groups of agents can achieve through coordinated action. However, these logics treat the system state as a monolithic whole, limiting support for modular verification. By contrast, resource logics such as Separation Logic, originally developed for sequential program verification, support compositional reasoning by partitioning state into disjoint regions, but they lack the means to express strategic interaction among agents. We present *Separating Coalition Logic* (SCL), a unified framework that combines strategic reasoning with resource-sensitive modularity. Built over ‘minimal’ interactive systems with local update rules and disjoint agent control, SCL supports both strategic guarantees and system decomposition along clean boundaries. A key feature is the *Strategic Frame Rule*, which lifts local properties into global guarantees even under adversarial conditions, enabling compositional proofs without reanalysis. We illustrate the framework via a case study in decentralized governance for Decentralized Autonomous Organizations, showing how SCL supports modular reasoning about both strategic interference and resource isolation.

Keywords: Logic · Substructural Logic · System Model · Strategic Verification · Decision-making · Separation · Interface · Blockchain

1 Introduction

Since von Neumann and Morgenstern’s classic *Theory of Games* [35], strategic reasoning has framed multi-agent systems as contests in which self-interested players reshape a shared environment. This intuition is crystallized in modal logic formalisms such as Coalition Logic [26], Alternating-time Temporal Logic and its many refinements [2], and Strategy Logic [15], where each agent is assigned a precise slice of a global transition structure and formulae declare exactly which coalitions (a subset of agents) can force which outcomes. A comparable conceptual breakthrough arose in modular program verification with *Separation Logic* [20,30], whose separating conjunction $*$ and *frame rule* treat the heap as a consumable resource: prove a property once for a disjoint fragment of state, and

the rule lifts that proof unchanged into any larger context. The separating conjunction and its frame rule give verification engineers a powerful routine: prove a property once for a small, self-contained slice of memory, and the rule hoists that proof unchanged into any compatible wider program context.

Modal logics of strategy such as Coalition Logic [26], Alternating-time Temporal Logic [2], and Strategy Logic [15] stipulate precisely *who can force what*, yet they treat the global state as an indivisible whole and therefore offer no support for re-using proofs when the system is extended or upgraded. By contrast, Separation Logic, and its extensions, excel at modular verification: its separating conjunction and frame rule allow a property to be proved once for a self-contained slice of state and then carry that guarantee unchanged into any compatible larger context [25,20,30,7]. What multi-agent systems really need is the best of both worlds, a logical apparatus whose notion of *strategy* is itself modular, so that fragments of a complex plan can be designed, verified and later composed without unintended interference, even in the presence of a resourceful adversary.

Decentralized Autonomous Organizations (DAOs) are blockchain-based systems in which collective decisions are governed by smart contracts and executed by distributed agents [6]. Verifying such infrastructures demands both strategic reasoning — for example, ‘can a majority of governors drain the treasury?’ — and resource-based guarantees — for example, ‘does a contract upgrade preserve collateral segregation?’. To capture both, we model the system as a *behavioural game structure* (cf. Section 3). This framework generalizes both the concurrent-game semantics used in strategic logics such as Coalition Logic and the resource semantics underlying separation-based verification, abstracting away from particular representations such as heaps to focus on compositionality and locality in system interaction (e.g., [27,10]). On this semantic base, we build *Separating Coalition Logic* (SCL), a modal language featuring strategic operators, two orthogonal separating conjunctions (over components and over agents), and a shared read-only interface guarded by an invariant. This structure supports compositional proofs that remain sound under adversarial interference. For instance, in the DAO setting, SCL enables concise verification of properties like ‘collateral never crosses the interface’ and ‘no attacker coalition can force a breach’, even under governance changes or smart contract upgrades.

The remainder of the paper motivates and reviews some of the existing formalisms (Section 2), formalizes the game structure (Section 3), gives the full SCL syntax and semantics (Section 4), and establishes the strategic frame rules (Section 5). Section 6 discusses DAOs in more detail, and uses SCL to specify some properties in a simplified yet concrete model of DAO in terms of game structures. Finally, in section 7 we explore future extensions and limitations of the current framework.

2 Background and motivation

As mentioned in Section 1, existing modal and game-theoretic logics for strategic ability, do not yet support modular reasoning about independent strategy components. Prior work (e.g., [3,18]) has observed aspects of substructural ‘resource-separating’ reasoning in temporal logics and reactive synthesis. We briefly discuss three key frameworks: Coalition Logic, ATL, and Strategy Logic.

Coalition Logic (CL) was introduced by Pauly as a modal logic describing the abilities of agent groups in games [26]. It provides an abstract game-theoretic modality $\langle A \rangle \phi$ (in some presentations) meaning ‘coalition A has a collective strategy to ensure ϕ .’ CL’s semantics, based on effectivity functions, gives a unifying game-theoretic view of modal operators. However, CL is limited in that it reasons about one coalition at a time achieving one goal. There is no direct way to say two disjoint coalitions can simultaneously achieve two separate objectives. Furthermore, CL lacks ‘temporal’ operators, so it cannot easily express ongoing or sequential strategic behaviour. It assumes a coalition’s strategy is executed in a single step or leads to an outcome in one move. This makes it insufficient for modelling richer multi-stage scenarios or reasoning about how strategies unfold over time.

Alternating-Time Temporal Logic (ATL) extends the effectivity semantics of Coalition Logic with temporal operators, enabling claims like $\langle\langle A \rangle\rangle F, \varphi$ (‘coalition A can ensure φ eventually’) [2,19]. Its semantics quantifies over all strategies for A against all counter-strategies of other agents, yielding strong worst-case guarantees. However, ATL lacks mechanisms for strategic modularity: it cannot express that two coalitions act independently on disjoint resources or reason about concurrent, non-interfering goals. Conjoined claims such as $\langle\langle A \rangle\rangle F, X \wedge \langle\langle B \rangle\rangle F, Y$ do not imply composable execution or separation of concerns. Nor can ATL distinguish antagonistic interference from benign independence, since it treats all non- A agents as adversaries. Moreover, standard ATL assumes memory-less perfect-information strategies, which restricts expressiveness and has led to costly extensions [4]. In sum, while ATL is powerful for global strategic reasoning, it lacks the structural tools needed for modular, resource-aware verification.

Strategy Logic (SL), introduced in [15], treats strategies as first-class objects and allows explicit quantification and assignment of strategies to agents. In SL, formulae such as $\exists s_A, \exists s_B. \varphi(s_A, s_B)$ mean ‘there exist specific strategies s_A for agent (or coalition) A and s_B for B such that formula φ holds’. This explicit game-theoretic approach subsumes ATL, and related modal strategic logics in expressiveness. However, these domains do not provide a fully natural setting in which substructural reasoning (and thereby a *frame rule*) can be leveraged in their entirety.

In contrast, our framework treats each game configuration as a first-class *resource*, allowing isolation, local verification, and composition along explicit interfaces. This substructural discipline reflects security-critical settings where adversaries are scoped by access to specific components. Classical strategies, which are functions from histories to actions, become *resource-sensitive*: agents act

only on their own write-blocks (the set of components whose behaviour they can guide), with all cross-influence channelled through a fixed interface. This mirrors established techniques in program and graph logics [7,9,30], where modularity arises from local reasoning and controlled composition. SCL *introduces* these principles into strategic reasoning, ensuring that behaviour composes soundly only through declared interfaces. The result supports modular verification in multi-agent systems such as DAOs, where both strategic coordination and resource isolation are essential. Section 3 formalizes the underlying semantics.

3 The Game Modelling Framework

Our atomic objects of study will be called *components* and *agents*. Ontologically, agents are entities that can shape or guide the dynamics of components' behaviours, and components are entities that make up a system that exhibit behaviours.

3.1 From System Models to Game Structures

The core machinery of our framework — components, behaviours and influence rules — is inherited from a more general notion of *system model* introduced in a recent work [13], and is reproduced below for completeness. We explain the formal meaning of some of the concepts necessary for the development of this paper in Section 3.2. For now, we explain the definition and use of the concept of a system model informally.

Definition 1 (System model). *For each $c \in \mathcal{C}$, define $\Delta_c := \{(f, f') \in F \times F \text{ such that } f'(c) = \mathcal{I}_c(f(c), (f(d))_{d \in \text{Inf}(c)})\}$. Here, for all $d \neq c$, $f'(d) = f(d)$. Let $\Delta_{\mathcal{I}} = \bigcup_{c \in \mathcal{C}} \Delta_c$. A system model $\mathcal{M}$ is a tuple $(\mathcal{C}, \mathcal{B}, \mathcal{I}, F, \Delta_{\mathcal{I}}, \Gamma)$, where F is the set of all possible configurations of the system given a set of components $\mathcal{C}$, a set of possible behaviours $\mathcal{B}$, and a family of rules $\mathcal{I}$ that governs the change in behaviour of the components. $\Gamma : \mathcal{P} \rightarrow 2^F$ is a valuation function that assigns a subset of the configurations to each atomic proposition from a set $\mathcal{P}$ of atomic propositions. $\square$*

A system model abstracts a dynamic system as a collection of interacting components, each capable of exhibiting distinct behaviours. The evolution of each component is governed by its own behaviour as well as its influence context: a fixed set of other components whose behaviours it observes. A configuration represents a global snapshot of the system, assigning a specific behaviour to every component. Informally, *influence rules* are 'local' behaviour update functions: each one reads the current behaviour of a component together with the behaviours of a small, fixed neighbourhood and deterministically produces that component's next behaviour. These updates collectively define a transition relation on configurations, yielding a graph of possible evolutions. Thus, a system model can be intuitively viewed as a directed graph over its configuration space:

vertices are configurations, and an edge $f \rightarrow f'$ exists whenever *one* component applies its influence rule while the rest of the state is frozen.

System models give us the minimal, component-centric view of dynamics. Example 1 (below) instantiates our behavioural-game framework with three components, explicit behaviour domains, and deterministic influence rules, illustrating autonomous evolution, context-dependent updates, and interface within a single ‘toy’ system. Informally, in this example, an *interface* is a distinguished subset of components whose behaviours serve as a boundary for interaction and coordination between otherwise disjoint subsystems. It is a subset of components such that for each interface component, its influence context (other components that influence its behaviour) is a subset of the interface itself.

Example 1 (From [13]). Let $\mathcal{C} = \{c_1, c_2, c_3\}$ and $\mathbb{B}(c_1) = \{b_{11}, b_{12}, b_{13}\}$, $\mathbb{B}(c_2) = \{b_{21}, b_{22}\}$, and, $\mathbb{B}(c_3) = \{b_{31}\}$. The influence contexts are $\text{Inf}(c_1) = \emptyset$, $\text{Inf}(c_2) = \{c_1\}$, $\text{Inf}(c_3) = \emptyset$. The influence rules are $\mathcal{I}_{c_1}(b_{11}) = b_{12}$, $\mathcal{I}_{c_1}(b_{12}) = b_{13}$, $\mathcal{I}_{c_1}(b_{13}) = b_{11}$, $\mathcal{I}_{c_2}(b_{21}, b_{12}) = b_{22}$, $\mathcal{I}_{c_2}(b_{21}, _) = b_{21}$, $\mathcal{I}_{c_2}(b_{22}, _) = b_{22}$, $\mathcal{I}_{c_3}(b_{31}) = b_{31}$. Thus c_1 cycles its behaviours autonomously, c_2 switches from b_{21} to the b_{22} *only when* c_1 behaves b_{12} , and c_3 is inert. Take the partition $C_1 = \{c_1, c_2\}$, $C_2 = \{c_1, c_3\}$, and, an interface, $I = C_1 \cap C_2 = \{c_1\}$. Refer to Figure 1. $\square$

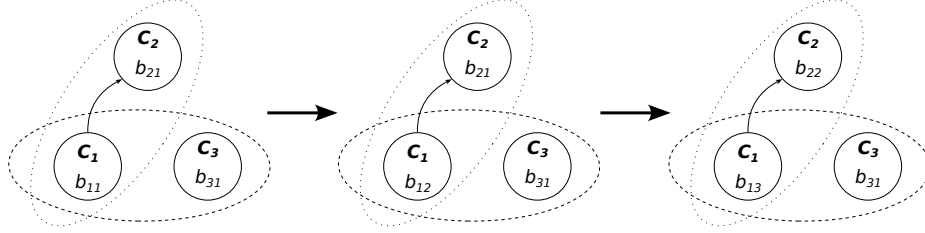


Fig. 1. A depiction of the system from Example 1. In the first evolution, component c_1 switches from $b_{11} \rightarrow b_{12}$. In the second, c_2 switches $b_{21} \rightarrow b_{22}$ and c_1 switches $b_{12} \rightarrow b_{13}$.

Game structures (Definition 4) add a set of agents, an agent partition of the components, and a one-step agent controlled evolution rule of components’ behaviours to system models. In this sense, game structures are not a new semantics, but a principled refinement of the system model tailored to multi-agent systems.

The game structure also incorporates a read-only interface I (Definition 7) which fits neatly into the system-model perspective: its components are not guided by any agents. In Definition 7, the additional interface invariant simply restricts the reachable portion of the set of all game configurations $\mathcal{F}$ to a ‘safe region’ $\text{Inv} \subseteq \mathcal{F}$, and is a design choice. The interested reader may also look at [16,17] for some related formalisms on interface and subsystems.

To turn Example 1 into a complete *game structure* in the sense of Definition 4, the behavioural core we already fixed (components, behaviour domains, influence

contexts, and influence rules) must be augmented with the *agency layer*: a finite set of agents $\mathcal{A}$ and a pairwise-disjoint write-block partition $(\{C_a\})_{a \in \mathcal{A}}$ that assigns each agent the components it may update. We then choose a read-only interface $I \subseteq \mathcal{C}$ together with an invariant Inv . Informally, in the presence of agency, interface can be thought of as read-only subset of components that agents can not guide but only observe. Their behaviours evolve autonomously without explicit agent control. An invariant, in this setting, is a condition that holds for those interface components in every configuration the game structure can ‘reach’, irrespective of which strategies agents adopt. Interfaces and invariants are discussed in Def. 7.

Example 2. Take $\mathcal{C} = \{c_1, c_2, c_3\}$ as before. Two agents, $\{a_1, a_2\}$ act on disjoint write-blocks, $C_{a_1} = \{c_1\}$ and $C_{a_2} = \{c_2\}$. The remaining component c_3 is a read-only interface $I = \{c_3\}$ whose value b_{31} never changes ($\text{Inv} = \{f \mid f(c_3) = b_{31}\}$). With these additions, and from Example 1 we obtain a game structure (cf. Def. 4), $\mathcal{S} = (\mathcal{C}, (\mathbb{B}(c)), (\text{Inf}(c)), (\mathcal{I}_c), \mathcal{A}, \{C_a\})$ for $\mathcal{A} = \{a_1, a_2\}$. $\square$

3.2 Game Structure

In what follows, we fix a *finite* set of components, $\mathcal{C} = \{c_1, \dots, c_m\}$. Each component $c \in \mathcal{C}$ has a non-empty set of *local behaviours* $\mathbb{B}(c) \neq \emptyset$. We freely use b, c for local behaviour and component variables, respectively. Finiteness keeps the presentation simple and suffices for most practical applications; extending to countable component sets is straightforward but notationally heavier.

Definition 2 (Configuration). *The global configuration space is the Cartesian product $\mathcal{F} := \prod_{c \in \mathcal{C}} \mathbb{B}(c)$. An element $f \in \mathcal{F}$ —written either f or $(f(c))_{c \in \mathcal{C}}$ —is a configuration. Intuitively $f(c)$ records the current behaviour of component c . More formally, a configuration is a function $f : \mathcal{C} \rightarrow \bigcup_{c \in \mathcal{C}} \mathbb{B}(c)$ such that $f(c) \in \mathbb{B}(c)$ for every $c \in \mathcal{C}$.* $\square$

For technical convenience we also view behaviours with an explicit component tag via the disjoint union $\bigcup_c \mathbb{B}(c)$, but the primary representation is the product $\mathcal{F}$.

Components evolve in discrete rounds according to *influence rules*. Each component c reads its own behaviour as well as the behaviours of a prescribed *influence context* $\text{Inf}(c) \subseteq \mathcal{C} \setminus \{c\}$, and updates its own behaviour deterministically. Formally, it is a function that, given the current behaviour of a component and the behaviours of some other selected components in the system, determines its next behaviour.

Definition 3 (Influence rule). *For every $c \in \mathcal{C}$ an influence rule is a total function $\mathcal{I}_c : \mathbb{B}(c) \times \prod_{d \in \text{Inf}(c)} \mathbb{B}(d) \rightarrow \mathbb{B}(c)$. We collect the family of all such rules in $\mathcal{I}_{\mathcal{C}} := (\mathcal{I}_c)_{c \in \mathcal{C}}$.* $\square$

The following definition captures the arena for SCL. Components evolve via fixed, local influence contexts, but agents may act by selecting behaviours for disjoint blocks of some subset of components, while the rest of the components evolve autonomously.

Definition 4 (Game structure). A game structure is the tuple $\mathcal{S} = (\mathcal{C}, \{\mathbb{B}(c)\}, \{\text{Inf}(c)\}, \mathcal{I}_c, \mathcal{A}, \{C_a\}_{a \in \mathcal{A}})$. Here, $\mathcal{C}$ is a finite set of components; $\mathbb{B}(c)$ is a non-empty behaviour domain of component c ; $\text{Inf}(c) \subseteq \mathcal{C} \setminus \{c\}$ is its influence context; $\mathcal{I}_c : \mathbb{B}(c) \times \prod_{d \in \text{Inf}(c)} \mathbb{B}(d) \rightarrow \mathbb{B}(c)$ are influence rules; $\mathcal{A}$ is a finite set of agents; $C_a \subseteq \mathcal{C}$, are pairwise disjoint subsets of $\mathcal{C}$ which refer to components controlled by a . Here, we reuse $\mathcal{F} = \prod_c \mathbb{B}(c)$ for its configuration set. $\square$

An agent a chooses the *next* behaviour of every component under its control, and its action profile is $\alpha_a \in \prod_{c \in C_a} \mathbb{B}(c)$. We write joint action profiles as $\alpha_C := \prod_{a \in C} \alpha_a$ and, $\alpha := \prod_{a \in \mathcal{A}} \alpha_a$.

Definition 5 (Transition relation). Given a configuration $f \in \mathcal{F}$, a coalition C , and a joint action profile α_C , the next configuration $\delta_{\alpha_C}(f) \in \mathcal{F}$ is defined component-wise by

$$\delta_{\alpha_C}(f)(c) := \begin{cases} \alpha_a(c) & \text{if } c \in C_a \text{ for some agent } a \in C, \\ \mathcal{I}_c(f(c), (f(d))_{d \in \text{Inf}(c)}) & \text{otherwise.} \end{cases}$$

Because every branch is total, each δ_α is a total function on $\mathcal{F}$. $\square$

Definition 6 (Strategy and play). Let $\mathcal{H}_{\text{fin}}$ denote the set of finite, non-empty sequences of configurations $(f^0, \dots, f^n)$ for $n \geq 0$. Here, for each $i < n$, there is a joint action profile α^i such that $f^{i+1} = \delta_{\alpha^i}(f^i)$. A (pure) strategy for agent $a \in \mathcal{A}$ is a total function $\sigma_a : \mathcal{H}_{\text{fin}} \rightarrow \alpha_a$, where $\alpha_a \in \prod_{c \in C_a} \mathbb{B}(c)$. A strategy profile for a coalition $C \subseteq \mathcal{A}$ is a tuple $\sigma_C := (\sigma_a)_{a \in C}$ drawn from the set $\text{Strat}_C := \prod_{a \in C} \sigma_a$. When $C = \mathcal{A}$, we drop the subscript and write σ . Given an initial configuration $f^0 \in \mathcal{F}$ and a strategy profile $\sigma_C \in \text{Strat}_C$, the unique (finite or infinite) play $\text{play}(f^0, \sigma_C)$ it induces is the sequence $(f^k)_{k \geq 0}$ such that for each $k \geq 0$, $f^{k+1} = \delta_{\alpha_C}(f^k)$. $\square$

Remark 1. For a strategy profile σ and an initial configuration f , we write $\text{play}(f, \sigma)_n$ to denote the n -th configuration of the unique play sequence. Let $f^0 \in \mathcal{F}$ be an initial configuration. A configuration $f \in \mathcal{F}$ is *reachable* from f^0 if there exist joint actions $\alpha^0, \dots, \alpha^{k-1}$ such that the iterated application $f^1 := \delta_{\alpha^0}(f^0), f^2 := \delta_{\alpha^1}(f^1), \dots, f^k := \delta_{\alpha^{k-1}}(f^{k-1})$ yields $f^k = f$. The set of such f is denoted $\text{Reach}(f^0)$. $\square$

These definitions clarify the two-tiered architecture of a game structure: the *resource layer*, comprising components $\mathcal{C}$ and their behaviour domains $\mathbb{B}$, and the *strategic layer*, consisting of agents $\mathcal{A}$, their write-blocks C_a , and history-dependent strategies. This aligns with the substructural perspective from Section 2, where components are treated as resources, agents as local controllers, and influence rules as fixed internal dynamics.

Definition 7 (Interface and invariant). Let $\mathcal{F}$ be the set of all configurations over a set of components $\mathcal{C}$ in a game structure. A subset $I \subseteq \mathcal{C}$ is an *interface* if $C_a \cap I = \emptyset$ for every $a \in \mathcal{A}$. The *default interface* is the largest such set, $I_{\text{def}} := \mathcal{C} \setminus \bigcup_{a \in \mathcal{A}} C_a$. Given an interface and a behaviour constraint $\mathcal{J} \subseteq \prod_{c \in I} \mathbb{B}(c)$, the

corresponding interface invariant is, $\text{Inv}_{\mathcal{J}} := \{f \in \mathcal{F} \mid \text{for all } \sigma \in \text{Strat}_{\mathcal{A}}, \text{ if } (f^n)_{n \geq 0} = \text{play}(f, \sigma), \text{ then } f^n \restriction_I \in \mathcal{J} \text{ for all } n\}$. Here, $f \restriction_I := (f(c))_{c \in I}$ is the tuple of interface-component behaviours in configuration f . $\square$

Interfaces designate globally visible but read-only components: no agent may assign behaviours to them. The set $\text{Inv}_{\mathcal{J}}$ collects exactly those start-configuration whose interface behaviour projection never leaves $\mathcal{J}$, regardless of how the agents act on the other components. This invariant satisfies two closure properties:

- *Safety*: if $f \in \text{Inv}_{\mathcal{J}}$, then every play starting from f remains within it; that is, $f^n \restriction_I \in \mathcal{J}$ for all n ; and,
- *Non-interference*: since the definition quantifies over all strategies $\sigma \in \text{Strat}_{\mathcal{A}}$, no coalition can force the interface outside $\mathcal{J}$.

As a result, the semantics of separating conjunction (Definition 10) and the strategic frame rule (Theorem 1) remain sound: every reachable configuration respects $\text{Inv}_{\mathcal{J}}$ by construction.

Remark 2. The present framework focuses on verifying properties of full games starting from a fixed initial configuration, but many real-world systems evolve through branching histories and dynamic reconfiguration. To accommodate such scenarios, one natural extension is to consider the *subgame separation* of a system: the set of all subgames rooted at reachable configurations. Each such subgame inherits the global structure but begins from a distinct intermediate state, reflecting how agents may re-enter or resume interaction in evolving environments. Our current framework serves as a foundational step toward a more expressive theory of dynamic, resource-sensitive game logic. $\square$

4 Separating Coalition Logic

In this section, we introduce the logical language $\mathcal{L}([\cdot], *, \otimes)$, designed to reason about both the strategic and structural dimensions of game models. The language builds upon a one-step coalition modality $[C]$ inherited from [26], which asserts that a given coalition C can enforce a property unilaterally, regardless of the moves made by others. We augment the logic with two substructural connectives — akin to the multiplicatives in, for example, [20,24], and rooted in the broader tradition of relevance logic (e.g., [28,29] in a vast literature) — which enable the structured decomposition of configurations. The first, $*$, expresses *component separation*: it decomposes the global configuration space into disjoint subregions, each supporting its own local property. The second, $\otimes$, captures *agent separation*: it asserts that disjoint coalitions of agents, operating on non-overlapping write-blocks, can simultaneously guarantee separate goals. These connectives generalize classical ideas from Separation Logic [30,24] and bring them into the strategic domain, enabling the modular composition of subgames through well-scoped interfaces. Together, the constructs of $\mathcal{L}([\cdot], *, \otimes)$ allow for local reasoning about global strategic behaviour, a capability essential for verifying decentralized systems with both adversarial interaction and non-interference requirements.

4.1 Syntax and Semantics

In this section, we introduce a language to reason about coalition abilities compositionally. Let $\mathcal{P}$ be a countable set of *atomic propositions* and $\mathcal{A}$ be a set of agents. The separating coalition language $\mathcal{L}([\cdot], *, \otimes)$ is given by the grammar

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \Box\varphi \mid [C]\varphi \mid \varphi * \psi \mid \varphi \otimes \psi$$

where, $p \in \mathcal{P}$, and, $C \subseteq \mathcal{A}$. We adopt the usual classical abbreviations: $\varphi \wedge \psi := \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi := \neg\varphi \vee \psi$, and $\Diamond\varphi := \neg\Box\neg\varphi$.

We read formulae of $\mathcal{L}([\cdot], *, \otimes)$ as follows: atomic propositions $p \in \mathcal{P}$ are read as ‘ p holds’; negation $\neg\varphi$ is read as ‘not φ ’; and disjunction $\varphi \vee \psi$ as ‘ φ or ψ ’; the modality $\Box\varphi$ is read as ‘ φ holds after every joint move’, and its dual $\Diamond\varphi$ as ‘ φ holds after some joint move’; the strategic operator $[C]\varphi$ is read as ‘coalition C has a joint move that ensures φ , regardless of what the others do’; the separating conjunction $\varphi_1 * \varphi_2$ is read as ‘ φ_1 and φ_2 hold independently on disjoint parts of the system (except for a shared interface)’; lastly, $\varphi_1 \otimes \varphi_2$ is read as ‘two disjoint coalitions can simultaneously enforce φ_1 and φ_2 without interfering’.

A joint strategy profile $\sigma \in \text{Strat}_{\mathcal{A}}$ induces a unique play $(f^n)_{n \geq 0}$ from any initial configuration $f^0 \in \mathcal{F}$, with each step evolving as $f^{n+1} := \delta_{\alpha^n}(f^n)$, where $\alpha^n := (\sigma_a(f^0), \dots, f^n)_{a \in \mathcal{A}}$.

Definition 8 (Game Model). A game model is a pair $\mathcal{G} = (\mathcal{S}, \Gamma)$, where $\mathcal{S}$ is a game structure as in Definition 4, and $\Gamma : \mathcal{P} \rightarrow 2^{\mathcal{F}}$ is a valuation function mapping each atomic proposition $p \in \mathcal{P}$ to the set of configurations in which p holds. $\square$

Definition 9 (Semantics). Let $\mathcal{L}([\cdot], *, \otimes)$ be the language of Separating Coalition Logic. Let $\text{PM} := \{(\mathcal{G}, f) \mid \mathcal{G} = (\mathcal{S}, \Gamma) \text{ is a game model and } f \in \mathcal{F}_{\mathcal{S}}\}$ be the class of all pointed models. Then the satisfaction relation is a subset $\models \subseteq \text{PM} \times \mathcal{L}([\cdot], *, \otimes)$, where $(\mathcal{G}, f) \models \varphi$ means that formula φ holds at configuration f in game model $\mathcal{G}$, defined as follows:

$$\begin{aligned} (\mathcal{G}, f) &\models p \text{ iff } f \in \Gamma(p) \\ (\mathcal{G}, f) &\models \neg\varphi \text{ iff } (\mathcal{G}, f) \not\models \varphi \\ (\mathcal{G}, f) &\models \varphi \vee \psi \text{ iff } (\mathcal{G}, f) \models \varphi \text{ or } (\mathcal{G}, f) \models \psi \\ (\mathcal{G}, f) &\models \Box\varphi \text{ iff for all } \alpha, (\mathcal{G}, \delta_{\alpha}(f)) \models \varphi \\ (\mathcal{G}, f) &\models [C]\varphi \text{ iff there exists } \alpha_C \text{ such that, for all } \alpha_{\bar{C}}, \\ &\quad (\mathcal{G}, \delta_{\alpha_C \cup \alpha_{\bar{C}}}(f)) \models \varphi \\ (\mathcal{G}, f) &\models \varphi_1 * \varphi_2 \text{ iff Definition 10 is satisfied} \\ (\mathcal{G}, f) &\models \varphi_1 \otimes \varphi_2 \text{ iff Definition 11 is satisfied} \end{aligned}$$

where Definition 10 and Definition 11 are the satisfaction clauses corresponding to $*$ and $\otimes$ respectively (see below), and, $C \cup \bar{C} = \mathcal{A}$, the set of all agents. $\square$

Definition 10 (Satisfaction for $*$). $(\mathcal{G}, f) \models \varphi_1 * \varphi_2$ iff there exists a partition $C_1 \cup C_2 \cup I = \mathcal{C}$ such that $f \in \text{Inv}$ and, where $A_1 := \{a \in \mathcal{A} \mid C_a \subseteq C_1\}$ and $A_2 := \{a \in \mathcal{A} \mid C_a \subseteq C_2\}$, there exist strategies $\sigma_{A_1} \in \text{Strat}(A_1)$ and $\sigma_{A_2} \in \text{Strat}(A_2)$ such that, for all counter-strategies $\tau_{A_2} \in \text{Strat}(A_2)$ and $\rho_{A_1} \in \text{Strat}(A_1)$,

- for all $n \geq 0$, $(\mathcal{G}, \text{play}(f, \sigma_{A_1} \cup \tau_{A_2})_n) \models \varphi_1$,
 - for all $n \geq 0$, $(\mathcal{G}, \text{play}(f, \rho_{A_1} \cup \sigma_{A_2})_n) \models \varphi_2$, and
 - for all $n \geq 0$, $\text{play}(f, \sigma_{A_1} \cup \tau_{A_2})_n \upharpoonright_I = \text{play}(f, \rho_{A_1} \cup \sigma_{A_2})_n \upharpoonright_I$,
-
- and, $\text{play}(f, \rho_{A_1} \cup \sigma_{A_2})_n \in \text{Inv}$

Definition 11 (Satisfaction for $\otimes$). $(\mathcal{G}, f) \models \varphi_1 \otimes \varphi_2$ iff there exist coalitions $A_1, A_2 \subseteq \mathcal{A}$ such that

- $A_1 \cap A_2 = \emptyset$, $W(A_1) \cap W(A_2) = \emptyset$, $f \in \text{Inv}$
 - there exist strategies $\sigma_{A_1} \in \text{Strat}(A_1)$ and $\sigma_{A_2} \in \text{Strat}(A_2)$ such that for all counter-strategies $\tau_{A_2} \in \text{Strat}(A_2)$ and $\rho_{A_1} \in \text{Strat}(A_1)$,
 - for all $n \geq 0$, $(\mathcal{G}, \text{play}(f, \sigma_{A_1} \cup \tau_{A_2})_n) \models \varphi_1$,
 - for all $n \geq 0$, $(\mathcal{G}, \text{play}(f, \rho_{A_1} \cup \sigma_{A_2})_n) \models \varphi_2$, and
 - for all $n \geq 0$, $\text{play}(f, \sigma_{A_1} \cup \tau_{A_2})_n \upharpoonright_I = \text{play}(f, \rho_{A_1} \cup \sigma_{A_2})_n \upharpoonright_I$,
-
- and, $\text{play}(f, \rho_{A_1} \cup \sigma_{A_2})_n \in \text{Inv}$.

In Definition 11, $W(A_1) := \bigcup_{a \in A_1} C_a$ denotes the write-block of A_1 , and similarly for A_2 . In both Definition 10 and 11, the two plays $\text{play}(f, \sigma_{A_1} \cup \tau_{A_2})_n$, and $\text{play}(f, \rho_{A_1} \cup \sigma_{A_2})_n$, must coincide on the read-only interface I for it is *usually false* that the *entire* configurations are equal. Since $\text{Inv} \subseteq \mathcal{F}$, the membership check must involve the full configuration, not its restriction. Including just one of the two plays is enough, since if $\text{play}(f, \rho_{A_1} \cup \sigma_{A_2})_n$ lies in Inv , then every other play that agrees with it on the interface lies in Inv as well.

Note that Definitions 10 and Definition 11 both follow the pattern of decomposing the meaning of the conjunctive ($*$ or $\otimes$) formula into the meanings of their component formulae together with a condition that regulates their combination. This is the standard pattern for substructural conjunctions.

5 Properties

5.1 Embedding Coalition Logic

Pauly's Coalition Logic (CL) [26] formalizes what a coalition can achieve *in one move*. Its semantics is based on an *effectivity function* E that lists, for every coalition C (a coalition is a subset of agents $\mathcal{A}$) and state s , the sets of outcomes C can enforce. We now show how the CL modality arises *directly* from our deterministic *game structure* (Definition 4).

Fix a game model $\mathcal{G} = (\mathcal{S}, I)$ with configuration space $\mathcal{F}$, and, joint-action transition $\delta_\alpha : \mathcal{F} \rightarrow \mathcal{F}$. Recall that for any coalition $C \subseteq \mathcal{A}$ we write $\alpha_C := \prod_{a \in C} \alpha_a$.

Definition 12 (Induced effectivity). For every coalition C and configuration f define $E_{\mathcal{G}}(C)(f) = \{X \subseteq \mathcal{F} \mid \exists \alpha_C \forall \alpha_{\bar{C}} \delta_{\alpha_C \cup \alpha_{\bar{C}}}(f) \in X\}$. Thus X is enforceable by C in one step iff C has a joint action that guarantees the successor lands in X regardless of the opponents. □

In CL, one writes $[C]\varphi$ for ‘ C can force φ in one move’. From Definition 12, we obtain $(\mathcal{G}, f) \models [C]_{\text{CL}} \varphi$ iff $\llbracket \varphi \rrbracket \in E_{\mathcal{G}}(C)(f)$, where $\llbracket \varphi \rrbracket := \{t \mid (\mathcal{G}, t) \models \varphi\}$. But, by construction of $E_{\mathcal{G}}$, the right-hand side is $\exists \alpha_C \forall \alpha_{\bar{C}} : (\mathcal{G}, \delta_{\alpha_C \cup \alpha_{\bar{C}}}(f)) \models \varphi$, which is *exactly* the semantic clause for our one-step coalition operator $[C]$ (Section 4.1). Hence, $[C]_{\text{CL}} \varphi \equiv [C]\varphi$ over $\mathcal{G}$.

5.2 Frame Rules

Fix a read-only *interface* $I \subseteq \mathcal{C}$ and the global invariant $\text{Inv} \subseteq \mathcal{F}$. For a formula ϕ , write $\text{atoms}(\phi)$ for the set of atoms occurring in ϕ .

Definition 13 (*I*-stable formula). Let $\mathcal{G}$ be a game model with configuration space F over a set of components $\mathcal{C}$, and let $I \subseteq \mathcal{C}$ be a fixed set of interface components. A formula $\phi \in \mathcal{L}([\cdot], *, \otimes)$ is said to be *I*-stable if, for all configurations $f, g \in \text{Inv}$, $f \upharpoonright_I = g \upharpoonright_I \implies ((\mathcal{G}, f) \models \phi \Leftrightarrow (\mathcal{G}, g) \models \phi)$. Here, $f \upharpoonright_I$ denotes the restriction of the configuration f to its values on components in I . $\square$

Lemma 1 (Boolean closure of *I*-stable formulae). If ϕ and χ are *I*-stable, then so are $\neg\phi$, and, $\phi \vee \chi$. $\square$

Proof. Immediate from Definition 13. $\square$

Theorem 1 (Strategic frame rule). Let $C \subseteq \mathcal{A}$ be a coalition of agents, and let $W(C) := \bigcup_{a \in C} C_a$ denote the write-block of C . Let $f \in \text{Inv}$ be a configuration, and let ϕ be an *I*-stable formula. If $(\mathcal{G}, f) \models [C]\psi$ and, $(\mathcal{G}, f) \models \phi$ then, $(\mathcal{G}, f) \models [C](\psi * \phi)$. $\square$

Proof. Let $\sigma_C \in \text{Strat}(C)$ be a strategy profile witnessing $[C]\psi$ at f . Let $\bar{C} := \mathcal{A} \setminus C$ be the complementary coalition. Since I is read-only, no strategy modifies the interface. Therefore, any play starting at f will preserve the interface projection and remain within Inv . By definition of $[C]\psi$, for any counter-strategy $\tau_{\bar{C}} \in \text{Strat}(\bar{C})$, the outcome play $g^0 g^1 \dots := \text{play}(f, \sigma_C \cup \tau_{\bar{C}})$ satisfies $(\mathcal{G}, g^n) \models \psi$ at each step.

Now fix any alternative strategy ρ_C for coalition C . Since ϕ is *I*-stable and ρ_C cannot modify I , the resulting play $h^0 h^1 \dots := \text{play}(f, \rho_C \cup \tau_{\bar{C}})$ agrees with the previous play on I . Hence, $(\mathcal{G}, h^n) \models \phi$ holds at every step by *I*-stability and the assumption that $(\mathcal{G}, f) \models \phi$. Thus all three conditions of the $*$ -conjunction are met, and, $(\mathcal{G}, f) \models [C](\psi * \phi)$. $\square$

The analogous ‘frame rule’ for $\otimes$ operator is stated in 2. It requires the notion of

Definition 14 (Strategic invariance). For a coalition $C \subseteq \mathcal{A}$, define

$$(\mathcal{G}, f) \models [C]\Box\varphi$$

iff there exists a strategy profile $\sigma_C \in \text{Strat}_C$ such that, for every counter-strategy $\rho_{\bar{C}} \in \text{Strat}_{\bar{C}}$, if

$$(f^n)_{n \geq 0} = \text{play}(f, \sigma_C \cup \rho_{\bar{C}}),$$

then

$$\forall n \geq 0, \quad (\mathcal{G}, f^n) \models \varphi.$$

We say that C can strategically maintain φ from f . $\square$

Theorem 2 (Strategic composition rule). *Let $\mathcal{G}$ be a game model, and $W(C)$ denote the write-block of coalition C . Let $C, D \subseteq \mathcal{A}$ be disjoint coalitions of agents such that, $W(C) \cap W(D) = \emptyset$. If $(\mathcal{G}, f) \models [C]\psi$ and, $(\mathcal{G}, f) \models [D]\phi$ then, $(\mathcal{G}, f) \models [C \cup D](\psi \otimes \phi)$. $\square$*

Proof. Let $\sigma_C \in \text{Strat}(C)$, and $\tau_D \in \text{Strat}(D)$ be strategies witnessing $[C]\psi$, and $[D]\phi$ respectively.

Let $\overline{C \cup D} := \mathcal{A} \setminus (C \cup D)$ be the complementary coalition. Define $\pi := \sigma_C \cup \tau_D \in \text{Strat}(C \cup D)$, a combined strategy profile. Consider any counter-strategy $\rho \in \text{Strat}(\overline{C \cup D})$, and let the resulting play be, $f^0 f^1 f^2 \dots := \text{play}(f, \pi \cup \rho)$.

Since $W(C) \cap W(D) = \emptyset$, the change in behaviours of components controlled by C and D are disjoint. No counter-agent can interfere with their respective components. Hence, the restriction $f^n \upharpoonright_{W(C)}$ satisfies ψ , by definition of $[C]\psi$, and the restriction $f^n \upharpoonright_{W(D)}$ satisfies ϕ , by definition of $[D]\phi$. Therefore, the combined strategy profile π guarantees $\psi \otimes \phi$, and we conclude, $(\mathcal{G}, f) \models [C \cup D](\psi \otimes \phi)$. $\square$

It is important to observe that the two frame rules for $*$ and $\otimes$ reflect distinct modes of composition. The frame rule for $*$ (Theorem 1) allows one to *extend* a known guarantee $[C]\psi$ with a passive, invariant fact ϕ that is stable under interface-preserving changes. No strategic contribution is needed from the rest of the system. In contrast, the composition rule for $\otimes$ requires that *both* coalitions, C and D , actively secure their own properties ψ and ϕ respectively. Only when those strategies are independently valid, and do not interfere, can we combine them. Thus, $*$ supports *modular framing* of ambient invariants, while $\otimes$ supports *parallel composition* of disjoint coalitional guarantees.

Once a formula includes strategic or separating connectives, namely $[\cdot]$, $*$, or $\otimes$, its truth can depend on ownership of write-blocks, the availability of joint moves, or quantification over disjoint substructures, all of which may involve components outside the interface I . Such formulae are therefore generally *not* invariant under changes to non-interface components. To ensure genuine interface-locality, we restrict I -stability to the fragment $\mathcal{L}_{st}$, which comprises Boolean combinations of atomic propositions evaluated solely on I :

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \quad \text{where } p \in \mathcal{P}, \text{ the set of atomic propositions}$$

These remain invariant under changes elsewhere in the system. Extending I -stability beyond this fragment is unsafe: connectives like $*$ or $\otimes$ quantify over disjoint subsets of components or agents that exclude I , so even a formula like $p * q$ (with $p, q \in I$) may change truth value if perturbing a non-interface component alters the admissible splits. Similarly, the truth of $[C]\varphi$ may shift if any component in $W(C)$ is affected, even with I unchanged. Accordingly, the frame rule (Theorem 1) applies only to I -stable state formulae, which suffice to express and preserve invariant interface properties.

6 Example: Decentralized Autonomous Organization

A *Decentralized Autonomous Organization* (DAO) is a blockchain-based governance structure where smart contracts encode rules and token holders vote on decisions without central oversight. A typical treasury-backed DAO comprises a **Governor** contract for queuing and executing proposals, a **Treasury** that disburses funds under economic constraints (e.g., at most 10% per transaction while maintaining 150% collateral), and an external **Oracle** that reports ETH/USD prices [36]. Strategic and resource concerns intersect in such settings: malicious coalitions may collude to drain funds, while Governor and Treasury operate on disjoint state but must preserve a shared safety invariant. See [22] for DAO governance structures, and [31] for regulatory perspectives.

This DAO setting offers an ideal use case for Separating Coalition Logic. Write-blocks correspond to contract storage: the Governor controls queue and timelock, the Treasury manages balance and debt. The price feed acts as a fixed interface which can not be changed by the agents. Within this structure, liveness (queued proposals execute), safety (collateral ratio $\geq 150\%$), and resilience (no attacker coalition without Treasury access can break the invariant) are all expressible as SCL formulae whose proofs can be established in a modular fashion (via $*$ and $\otimes$ connectives). For further background on the governance–treasury split, see [1], and for attacker coalitions and oracle manipulation, see [34].

6.1 Verifying a Collateral-backed DAO.

We illustrate the expressiveness of *Separating Coalition Logic* (SCL) by formalizing a simplified model of a DAO, such as those used for collateral-backed governance. This is used to demonstrate how SCL’s separating conjunctions ($*$ and $\otimes$) express component- and agent-level modularity, and, to show how the strategic modality $[C]\varphi$ captures the capabilities of both honest and adversarial coalitions.

To illustrate how *Separating Coalition Logic* (SCL) captures both resource separation and strategic reasoning, we consider a simplified model of a collateral-backed **Decentralized Autonomous Organization** (DAO). In our simplified example, the DAO is described by the following set of components, $\mathcal{C} = \{\text{qLen}, \text{timelock}, \text{balance}, \text{liab}, \text{ethPrice}\}$. Each component $c \in \mathcal{C}$ is associated with a local behaviour space $\mathbb{B}(c)$, and together they model the observable state of the system. Each component is associated with a specific operational role:

qLen	: length of the Governor’s proposal queue
timelock	: seconds remaining before next proposal executes
balance	: ETH collateral held in the Treasury
liab	: outstanding debt (denominated in USD)
ethPrice	: ETH/USD market rate supplied by the Oracle

The DAO operates by allowing token-holders to propose and vote on actions via the *Governor*, which manages a queue of proposals and a time-delay mechanism (qLen, timelock). Once a proposal’s timelock expires, it can execute changes

to the system state, most notably, transferring funds from the *Treasury*, which is governed by the parameters **balance** and **liab**. An external *Oracle* feeds the current ETH/USD price into the system via **ethPrice**, and the entire system is considered secure as long as the collateral ratio $\text{balance} \cdot \text{ethPrice} \geq 1.5 \cdot \text{liab}$ is maintained. This simplified abstraction captures the key resource dependencies and strategic controls needed to express and verify governance, safety, and adversarial properties in the DAO.

Each component $c \in \mathcal{C}$ is associated with a local behaviour domain,

$$\begin{aligned} \mathbb{B}(\mathbf{qLen}) &= \{0, 1, \dots, 10\} & \mathbb{B}(\mathbf{timelock}) &= \{0, 1, \dots, 172800\} \\ \mathbb{B}(\mathbf{balance}) &= \mathbb{R}_{\geq 0} & \mathbb{B}(\mathbf{liab}) &= \mathbb{R}_{\geq 0} \\ \mathbb{B}(\mathbf{ethPrice}) &= \mathbb{R}_{> 0} \end{aligned}$$

A *configuration* $f \in F$ is a function $f : \mathcal{C} \rightarrow \bigcup_{c \in \mathcal{C}} \mathbb{B}(c)$ with $f(c) \in \mathbb{B}(c)$ for each $c \in \mathcal{C}$. The component space $\mathcal{C}$ is partitioned among agents as follows:

$$\begin{aligned} \text{GovCtrl} &= \{\mathbf{qLen}, \mathbf{timelock}\} & \text{TreCtrl} &= \{\mathbf{balance}, \mathbf{liab}\} \\ \text{Oracle} &= \emptyset \quad (\text{read-only}) & \text{Attacker} &= \emptyset \quad (\text{off-chain only}) \end{aligned}$$

We list Oracle and Attacker among the agents to reflect their strategic or observational role, even though they control no components; this allows reasoning about adversarial goals, external influence, or non-interference conditions involving them.

Each component evolves via a deterministic influence rule $\mathcal{I}_c$, which updates its value based on a local context:

$$\begin{aligned} \mathcal{I}_{\text{balance}}(b, x) &= b - x & \text{for } x \leq 0.1 \cdot b \quad (\text{bounded payment}) \\ \mathcal{I}_{\text{timelock}}(t) &= \max(t - 12, 0) & (\text{epoch countdown}) \\ \mathcal{I}_{\text{ethPrice}}(p) &= p & (\text{external oracle; constant over one step}) \end{aligned}$$

Some examples of agent induced transitions are given below. Suppose the current configuration f satisfies: $f(\mathbf{balance}) = 100$, and, $f(\mathbf{timelock}) = 24$. Agent *TreCtrl* proposes $\text{pay}(x)$ with $x = 5 \leq 0.1 \cdot 100$, and then, $f'(\mathbf{balance}) = \mathcal{I}_{\text{balance}}(100, 5) = 95$. Similarly, on a ‘system tick’, the governor-controlled **timelock** decrements, $f'(\mathbf{timelock}) = \mathcal{I}_{\text{timelock}}(24) = 12$. Oracle-controlled **ethPrice** remains unchanged unless externally perturbed, $f'(\mathbf{ethPrice}) = \mathcal{I}_{\text{ethPrice}}(p) = p$.

To ensure safety, we enforce an interface invariant for over-collateralization: $\text{Inv} = \{f \in F \mid f(\mathbf{balance}) \cdot f(\mathbf{ethPrice}) \geq 1.5 \cdot f(\mathbf{liab})\}$. This expresses that the DAO maintains at least 150% collateralization with respect to outstanding liabilities, and enables SCL to verify compositional safety guarantees in the presence of strategic coalitions. Together, these properties allow us to reason about the compositional security of the DAO under varying strategic assumptions.

Remark 3. While we have so far used quantitative behaviours such as token balances or collateral ratios, the same framework readily extends to settings where

behaviours are qualitative; for instance, status flags, policy regimes, or operational classifications. This makes it applicable to domains like economic security and military command-and-control, where components may encode regulatory compliance states or mission-critical statuses (e.g., ‘under audit’, ‘compromised’, ‘active’). In such contexts, influence rules capture institutional or procedural dynamics, and agent strategies model the actions of regulators, command units, or adversaries. SCL enables compositional reasoning by isolating subgames (e.g., market sectors or subsystems), enforcing coordination through interfaces, and ensuring that strategic interference is contained. Its strategic frame rule ensures that local guarantees, such as regulatory conformance or operational integrity, can be lifted compositionally even in the presence of adversarial scenarios.

6.2 DAO Objectives as SCL Formulae

Some DAO governance objectives, such as liveness and safety constraints, can be modularly expressed in the logic SCL:

Liveness of governance: When the proposal queue is non-empty and the timelock has expired, the *GovCtrl* agent is capable of executing the head proposal: $\varphi_{gov} = (\text{qLen} > 0 \wedge \text{timelock} = 0) \rightarrow [\text{GovCtrl}](\text{qLen} = 0 \wedge \text{timelock} > 0)$.

Treasury safety: The *TreCtrl* agent may initiate payments, but only up to ten percent of the current balance. This safety condition is enforced by (Δ refers to difference): $\varphi_{tre} = [\text{TreCtrl}] (\Delta \text{balance} \geq -0.1 \cdot \text{balance})$

Global collateral ratio: We encode the interface predicate $\phi_{safe} = \text{balance} \cdot \text{ethPrice} \geq 1.5 \cdot \text{liab}$ which is *I*-stable and enforced as an invariant throughout. This condition ensures that the DAO remains at least 150% collateralized in all reachable configurations.

The system designer can express the local objectives of the Governor and the Treasury as a *resource-separating* specification: $[\text{GovCtrl}] \varphi_{gov} * [\text{TreCtrl}] \varphi_{tre}$. Since $C_{gov} \cap C_{tre} = \emptyset$ and both agents leave the oracle value *ethPrice* unchanged, the semantic conditions of the separating conjunction $*$ are satisfied.

By applying the *strategic frame rule* with the interface invariant ϕ_{safe} , we obtain the global guarantee, $[\text{GovCtrl}](\varphi_{gov} * \phi_{safe}) \wedge [\text{TreCtrl}](\varphi_{tre} * \phi_{safe})$. Informally, this states that the Governor can process the proposal queue *while* maintaining the collateral ratio, and the Treasury can safely disburse funds *while* ensuring the same invariant without re-analysing the correctness of the other subsystem.

We can also specify a threat model using *coalition separation*. Suppose the adversary compromises the governance controller, but not the Treasury. Let $C_{adv} = \{\text{GovCtrl}, \text{Attacker}\}$ be the coalition under adversarial control. We want to assert that this coalition *cannot* violate the interface invariant without also gaining control over the Treasury.

This is captured by the formula, $\chi_{adv} = \neg[C_{adv}] \neg\phi_{safe}$, which guarantees that, even under malicious coordination between governance and the network layer, the system remains collateral-safe as long as *TreCtrl* operates correctly. Using the *agent-separating* connective we capture the assumption that Treasury

remains honest while the Governor may collude,

$$\underbrace{\neg [C_{adv}] \neg \phi_{safe}}_{\text{attacker fails}} \otimes \underbrace{[TreCtrl] \phi_{safe}}_{\text{honest Treasury defends}}$$

Because the write-blocks of the two coalitions are disjoint and both sides respect the interface, the formula is true in the initial configuration and remains so. Hence SCL formally records the intuitive ‘two-man rule’: without the Treasury’s private key the Governor cannot drain enough funds to breach the collateral threshold.

The DAO example demonstrates how SCL supports modular reasoning along both structural and strategic dimensions. The connective $*$ separates the Governor and Treasury contracts, enabling each to be verified independently. The connective $\otimes$ distinguishes honest from adversarial coalitions, allowing threat models to be analysed separately from functional correctness. The result is a concise yet faithful model of a real-world DAO whose key safety and liveness claims can be checked compositionally.

7 Discussions

Game-theoretic techniques have long informed formal verification, notably in model checking, controller synthesis, and verification of open systems [5,33], where the core question is whether a strategy can guarantee a specification against all environmental behaviours. Such strategies effectively carve out substructures where the goal holds, making reasoning about strategic submodels central to verification. While our framework is currently qualitative, the modality $[C]\varphi$ states that coalition C can enforce φ without reference to quantitative preferences, it naturally invites quantitative extensions. One direction is to equip agents with utility functions $u_a : F \rightarrow \mathbb{R}$ and introduce modal forms like $[C]_{\geq k}\varphi$, asserting that C can jointly ensure both φ and a minimum payoff. Similar ideas appear in quantitative ATL and Strategy Logic, but integrating them with substructural connectives raises subtle issues: how should utilities decompose across $*$ and $\otimes$? We conjecture that utilities can be blended into Separating Coalition Logic without breaking its modular ‘frame’ discipline, provided the numerical objectives themselves are interface-stable. Concretely, a payoff function $u : F \rightarrow \mathbb{R}$ can be called interface-stable when its value depends only on the read-only interface I : if two configurations agree on I they yield the same utility. Because no agent is ever allowed to write to I , every strategy profile, honest or adversarial, preserves the interface projection.

The key semantic ideas underlying SCL were seeded in earlier work one of the authors, and colleagues. The notion of a read-only *interface slice*, through which subsystems can observe but not interfere, was first formalized in [8]. Coalition-level strategic reasoning appeared in [11], which studied how stakeholder groups adopt privacy-enhancing technologies, and is conceptually captured by the SCL modality $[C]\varphi$. The separation discipline that underpins modularity in SCL

draws on the policy composition framework of [12], where state partitioning and local invariants played a central role.

The modality $[C]\varphi$ retains the classical coalition semantics of Pauly [26]: coalition C has a single joint move that ensures φ against every possible counter-move. However, within our behavioural game structures where component and agent separation are built in, and interactions are mediated through an immutable interface. This quantification composes modularly with frame rules (cf. Theorem 1) under local invariants. Such compositionality is absent in traditional Coalition Logic and is central to SCL’s support for qualitative security guarantees.

An instructive analogy arises with Stackelberg security games [32], where a defender commits to a mixed patrol strategy anticipating an adaptive attacker. The leader’s guaranteed coverage maps conceptually to an SCL formula $[Def]\varphi$, with patrol sectors corresponding to disjoint resource parcels governed by $*$ or $\otimes$. The contrast lies in foundations: Stackelberg models rely on Bayesian assumptions and expected utility maximization, whereas SCL operates deterministically, emphasizing invariant reachability and strategic non-interference without presupposing preferences or probabilistic priors.

7.1 Future Work

In our framework, we deliberately refrain from imposing preference relations or utilities directly over component behaviours. Strategies in Separating Coalition Logic (SCL) are treated purely as intensional mappings from histories to actions without reference to agents’ goals or outcomes. Introducing explicit preferences or utilities would shift semantics from an intensional account of strategic enforceability to a teleological account based on rationality and payoff maximization. This would compromise the essential locality and modularity upon which resource-sensitive separation and interface stability depend. However, incorporating preferences as a secondary semantic layer remains a valuable direction for future extensions.

With quantitative *utilities* in place, Separating Coalition Logic (SCL) could move beyond qualitative safety and express fine-grained *budget constraints* (‘the treasury always keeps at least X collateral’), *cost-bounded attacks* (‘the adversary must burn k tokens before the guard fails’), and even *incentive-compatible* DAO voting. A natural extension is to enrich the two separating connectives with *additive* and *min-max* frame principles: whenever two sub-games are resource-separated, the payoff a coalition secures in the whole game should decompose into the sum, or the minimum, depending on the objective, of the payoffs in the parts, subject to the shared interface invariant. Such quantitative frame rules would enable compositional reasoning about gas costs, slashing penalties, and token-weighted governance, while aligning SCL with SMT-based tool-chains already used for numerical objectives [14,23].

The current framework has two notable limitations. First, the one-step strategy operator yields only single-shot guarantees; expressing long-run properties such as liveness under adversarial conditions still requires external fixed-point

machinery like the μ -calculus encodings of parity games [21]. Second, the interface is assumed correct by fiat (e.g., trusted price oracles in DAOs), excluding oracle-manipulation attacks from the model. Future work aims to address both issues: by integrating lightweight temporal fixed points that preserve modular reasoning, and by promoting the interface to a first-class, read-only agent whose integrity can be verified within the strategic logic itself. Taken together, these advances will move SCL closer to a self-contained framework capable of end-to-end verification for strategic systems whose correctness hinges on both adversarial behaviour and resource separation.

Acknowledgements

Chakraborty is supported by UKRI through the Centre for Doctoral Training in Cybersecurity at UCL. Caulfield and Pym acknowledge the partial support of UKRI Research Grants EP/R006865/1 and EP/S013008/1.

References

1. a16zcrypto: A Legal Framework for Decentralized Autonomous Organizations (2022), <https://api.a16zcrypto.com/wp-content/uploads/2022/06/dao-legal-framework-part-1.pdf>
2. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. *J. ACM* **49**(5), 672–713 (2002). <https://doi.org/10.1145/585265.585270>
3. Benerecetti, M., Mogavero, F., Murano, A.: Reasoning about substructures and games. *ACM Trans. Comput. Logic* **16**(3) (2015). <https://doi.org/10.1145/2757286>
4. Beutner, R., Finkbeiner, B.: On alternating-time temporal logic, hyperproperties, and strategy sharing. In: *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence* (2024). <https://doi.org/10.1609/aaai.v38i16.29679>
5. Bloem, R., Chatterjee, K., Jobstmann, B.: *Graph Games and Reactive Synthesis*, pp. 921–962. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-10575-8_27
6. Bodellini, M., Zhang, K., Enyi, J.: Decentralized autonomous organizations (daos). In: *Financial Technology and Digital Commercial Law*. Oxford University Press (02 2025). <https://doi.org/10.1093/law/9780192868763.003.0013>
7. Brookes, S.: A semantics for concurrent separation logic. In: Gardner, P., Yoshida, N. (eds.) *CONCUR 2004*. pp. 16–34. Springer (2004)
8. Bujorianu, Manuela and Caulfield, Tristan and Ilau, Marius-Constantin, and Pym, David: Interfaces in ecosystems: Concepts, form, and implementation. In: *Simulation Tools and Techniques*. pp. 27–47 (2025)
9. Calcagno, C., Gardner, P., Zarfaty, U.: Context logic and tree update. *SIGPLAN Not.* **40**(1), 271–282 (2005). <https://doi.org/10.1145/1047659.1040328>
10. Caulfield, T., Ilau, M.C., Pym, D.: Engineering Ecosystem Models: Semantics and Pragmatics. In: *Simulation Tools and Techniques*. pp. 236–258. Springer (2022)
11. Caulfield, T., Ioannidis, C., Pym, D.: On the adoption of privacy-enhancing technologies. In: *7th International Conference on Decision and Game Theory for Security*. pp. 175–194 (2016). https://doi.org/10.1007/978-3-319-47413-7_11

12. Caulfield, T., Pym, D.: Modelling and simulating systems security policy. *EAI Endorsed Transactions on Security and Safety* **3**(8) (8 2015). <https://doi.org/10.4108/eai.24-8-2015.2260765>
13. Chakraborty, P., Caulfield, T., Pym, D.: Causality and decision-making: A logical framework for systems and security modelling (2025), <https://arxiv.org/abs/2508.01758>, Accessed 10 August 2025
14. Chatterjee, K., Doyen, L.: Perfect-information stochastic games with generalized mean-payoff objectives. In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*. pp. 247–256. *LICS '16* (2016). <https://doi.org/10.1145/2933575.2934513>
15. Chatterjee, K., Henzinger, T.A., Piterman, N.: Strategy logic. In: *CONCUR 2007 – Concurrency Theory*. pp. 59–73 (2007)
16. Galmiche, D., Lang, T., Méry, D., Pym, D.: Bifurcation Logic: Separation Through Ordering, To appear, *Proc. TARK XX*. *EPTCS* 2025, Manuscript: <https://www.cantab.net/users/david.pym/current.html>
17. Galmiche, D., Lang, T., Pym, D.: Minimalistic System Modelling: Behaviours, Interfaces, and Local Reasoning. In: *Proc 16th EAI SIMUtools*, Springer, 2024 (2024), <https://doi.org/10.48550/arXiv.2401.16109>, Accessed 9 June 2025
18. Gutierrez, J., Harrenstein, P., Wooldridge, M.: Reasoning about equilibria in game-like concurrent systems. In: *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning*. pp. 408–417 (2014)
19. Herzig, A., Lorini, E., Walther, D.: Reasoning about actions meets strategic logics. In: Grossi, D., Roy, O., Huang, H. (eds.) *Logic, Rationality, and Interaction*. pp. 162–175. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
20. Ishtiaq, S.S., O’Hearn, P.W.: BI as an assertion language for mutable data structures. In: *Proc. 28th ACM Symp. on Principles of Programming Languages*. pp. 14–26. ACM (2001). <https://doi.org/10.1145/360204.375719>
21. Karelavic, B., Zielonka, W.: Nearest fixed points and concurrent priority games. In: *Fundamentals of Computation Theory*. pp. 381–393 (2015)
22. Law, A.W.: The Rise of Decentralized Autonomous Organizations: Opportunities and Challenges. *Stanford Journal of Blockchain Law & Policy* (jun 30 2021), <https://stanford-jblp.pubpub.org/pub/rise-of-daos>
23. Lu, Z., Siemer, S., Jha, P., Day, J., Manea, F., Ganesh, V.: Layered and staged monte carlo tree search for smt strategy synthesis. In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence. IJCAI '24* (2024). <https://doi.org/10.24963/ijcai.2024/211>
24. O’Hearn, P.W., Pym, D.J.: The Logic of Bunched Implications. *Bulletin of Symbolic Logic* **5**(2), 215–244 (1999). <https://doi.org/10.2307/421090>
25. O’Hearn, P.W., Reynolds, J.C., Yang, H.: Local reasoning about programs that alter data structures. In: *Proceedings of the 15th International Workshop on Computer Science Logic*. pp. 1–19. *CSL '01*, Springer-Verlag, Berlin, Heidelberg (2001)
26. Pauly, M.: A modal logic for coalitional power in games. *Journal of Logic and Computation* **12** (02 2002). <https://doi.org/10.1093/logcom/12.1.149>
27. Pym, D.: Resource semantics: logic as a modelling technology. *ACM SIGLOG News* **6**(2), 5–41 (2019), <https://doi.org/10.1145/3326938.3326940>
28. Read, S.: *Relevant Logic*. Blackwell (1988)
29. Restall, G.: *An Introduction to Substructural Logics* (1st ed.). Routledge (2000), <https://doi.org/10.4324/9780203016244>
30. Reynolds, J.C.: Separation Logic: A Logic for Shared Mutable Data Structures. In: *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*. pp. 55–74. *LICS '02*, IEEE Computer Society (2002)

31. Securities and Exchange Commission: Report of investigation pursuant to section 21(a) of the securities exchange act of 1934: The dao, <https://www.sec.gov/files/litigation/investreport/34-81207.pdf>
32. Sinha, A., Fang, F., An, B., Kiekintveld, C., Tambe, M.: Stackelberg security games: looking beyond a decade of success. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 5494–5501. AAAI Press (2018)
33. Stevens, P., Stirling, C.: Practical model-checking using games. In: Proceedings of the 4th International Conference on Tools and Algorithms for Construction and Analysis of Systems. pp. 85–101. TACAS '98 (1998)
34. United States Department of The Treasury: Illicit Finance Risk Assessment of Decentralized Finance (2023), <https://home.treasury.gov/system/files/136/DeFi-Risk-Full-Review.pdf>
35. von Neumann, John and Morgenstern, Oskar: Theory of Games and Economic Behavior. Princeton University Press (1944)
36. Wang, S., Ding, W., Li, J., Yuan, Y., Ouyang, L., Wang, F.Y.: Decentralized autonomous organizations: Concept, model, and applications. IEEE Transactions on Computational Social Systems **6**, 870–878 (2019). <https://doi.org/10.1109/TCSS.2019.2938190>