

# BRE Arguments

---

This memorandum provides a comprehensive, data-driven rebuttal to the argument that a dedicated Rules Engine (RE) is unnecessary for the K12 modernization project and refutes the architectural feasibility of prioritizing the Enrollment Builder (EB) and unnecessary custom query tools over core systems and compliance requirements.

## Rebuttal: Necessity of a Centralized Rules Engine for K12 Modernization

The assertion that this mission-critical K12 system does not require a Rules Engine is unsound and contradicts documented statutory and financial requirements. Implementing a dedicated RE is not an option but an **essential architectural prerequisite** for achieving **100% regulatory compliance**, system scalability, and consistency in financial transactions, especially given the project's hard deadline of May 1, 2026.

### I. Mandate for Policy Externalization Due to Regulatory and Financial Complexity

The K12 scholarship system, spanning the Opportunity Scholarship (OS) and Education Student Accounts (ESA+) programs, is defined by numerous highly complex, statutory, and volatile rules that must be enforced consistently across all portals and financial workflows.

A dedicated **Rules Engine (RE)** is necessary to serve as a **Policy Service** for the following critical financial, eligibility, and compliance processes, mitigating the risk of embedding this logic in disparate application code:

#### A. Financial and Disbursement Logic

1. **Dual Award Ordering:** The system must strictly apply OS funds first to tuition and fees, before allocating any remaining ESA+ funds, which then move to ClassWallet for allowable expenses. This orchestration requires precise rule execution during the payment cycle.
2. **Minimum Spend Enforcement:** The RE must verify that ESA+ students spend **at least \$1,000 per school year** on allowable expenses (in core academic subjects) to remain eligible for renewal. Failure to meet this invariant must result in automatic non-renewal processing.
3. **Rollover Calculation and Caps:** For higher award (\$17,000) ESA+ recipients, the RE must calculate and enforce the rollover eligibility, capping the annual rollover at **\$4,500** and ensuring the student's cumulative balance does not exceed **\$30,000**.
4. **Transaction Fees:** The RE must enforce transaction fee limits charged to the account holder, ensuring the fee does not exceed 2.5% of the cost of the item or service.

#### B. Allowable Expense and Purchasing Compliance

1. **Allowable Expense Categorization:** Every **PurchaseRequest** submitted via ClassWallet must be validated against a dynamic list of allowable and prohibited expenses (e.g., household items, entertainment-only products).
2. **Educational Technology Constraints:** The RE must enforce complex timing constraints for technology accessories, which, if not bundled, must be purchased **within 30 days** of the main device order, and must enforce frequency limits (e.g., purchase of major devices only once every 3 years).

3. **Service Provider Vetting:** The RE must ensure that service-based expenses (tutoring, therapy) are only approved if the referenced `ProviderId` has an `enrollmentStatus == enrolled` and holds current credentials as validated by the **Provider Enrollment Context**.

## C. Eligibility and Award Determination

1. **OS Income Calculation:** The RE must execute complex **Income Calculation Rules** in accordance with Federal Free and Reduced-Price Lunch regulations, distinguishing between reportable income (gross wages, self-employment net income, annuities) and excludable income (Combat Pay, foster payments). This calculation dictates the precise **Award Tier** and amount received.
2. **Eligibility Lifecycle:** The RE is vital for checking renewal invariants, such as triggering the mandatory **3-year re-evaluation cycle** for disability documentation for ESA+ students.

## II. Architectural Imperatives and Risk Mitigation

The complexity and the high-risk nature of the K12 domain necessitate the architectural pattern of separating policy from execution via a Rules Engine.

1. **Consistency and Testability:** By centralizing these rules (which number far greater than standard application business logic), the system gains **consistency** (all transaction and eligibility paths rely on a single, versioned policy) and **testability** (rules can be unit-tested without coupling them to the persistence or API layers).
2. **Policy Versioning:** Because program rules and statutory requirements often change annually, the implementation plan specifically mandates a Rules Engine for **Policy Versioning** by program year. This is essential for managing compliance across different fiscal years without requiring extensive code changes (code churn).
3. **Workflow Automation vs. Ad-Hoc Queries:** The suggestion to rely on administrators using a query builder tool to generate dynamic recipient lists for *core communications* is fundamentally flawed. Automated, event-driven processes (such as deadline reminders, suspension notices, and payment notifications) must be tied directly to **Rules Engine** evaluations and **Domain Events** (e.g., `MinimumSpendFailed`, `AwardSuspended`, `PurchaseRejected(code)`). Placing a manual human step (using a query builder) into time-critical, high-volume automated communication workflows introduces unacceptable risk of error, delay, and non-compliance. The centralized **Communication Center** and **Rules Engine** in Phase 2 are designed to automate these processes, not enable ad-hoc query selection for critical tasks.

## III. Prioritization and Rebuttal of High-Risk, Low-Value Tools

The effort required to implement the Rules Engine, alongside other cross-cutting services (like the **Communication Center** and **Messaging Center**), is a necessary commitment that supersedes the development of low-ROI features.

The proposed schedule places **Rules Engine Integration** in Phase 2 (Weeks 11-12) as essential cross-cutting infrastructure. This implementation is only possible if engineering resources are immediately redirected from high-risk endeavors.

1. **Enrollment Builder (EB) Technical Debt:** The EB is an **unproven architectural component** lacking a formal state machine and a functional data mapping layer. It creates an unacceptable risk of data loss and corruption due to its dynamic data structure attempting to map to a static core schema.

2. **EB Schedule Failure:** Continuing the EB effort guarantees missing the **May 1, 2026 hard deadline** by six months or more. The plan explicitly states that prioritizing core features using **Angular reactive forms** (Option B) is the *only viable path* to meet the commitment.
3. **EB Low ROI:** The primary benefit claimed for the EB (reducing developer effort for legislative form changes) is undermined by the fact that such changes are rare (estimated at approximately 40 hours per year). The monetary cost (\$\$180K-\$240K\$) and schedule delay caused by the EB far outweigh this minimal annual saving.

**Conclusion:** The financial, compliance, and statutory complexity of the K12 scholarship programs mandates a dedicated, versioned Rules Engine as core platform infrastructure. Redirecting resources away from the high-risk, low-value Enrollment Builder (EB) is the immediate action required to staff the implementation of the essential Rules Engine and other critical cross-cutting services needed to meet the May 1, 2026 deadline.