GENETRIX

MIGRATION ARCHITECTURE SERIES

The Developer's Translation Matrix: Legacy SFMC to Core (MCG)

> **Context:** Legacy Marketing Cloud Engagement (MCE) relied on SSJS and AMPScript evaluated strictly at the time of send. Marketing Cloud on Core (MCG) enforces strict platform limits to protect CRM processing power. Complex scripting must shift "up" the funnel into Salesforce Flow, Apex Invocables, or Data Cloud Batch Transforms. Use this 10-point matrix to re-architect your codebase for native Core compliance.

1. CRUD & Database Operations

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
`LookupRows()`	**Flow:** Get Records **Apex:** `[SELECT Id FROM Obj__c]`	**The Trap:** Querying massive datasets at send-time will hit SOQL governor limits. **The Fix:** Pre-join complex relational data in Data Cloud DMOs, or traverse native parent/child Object relationships via Formula fields before the Flow runs.
`InsertData() / .Rows.Add()`	**Flow:** Create Records **Apex:** `insert recordList;`	Flat Data Extensions are dead. You are now inserting CRM records (Standard/Custom Objects). You must adhere to strict platform schema, validation rules, and required fields.
`ClaimRow()` (Coupons)	**Apex:** `[SELECT... FOR UPDATE]`	To prevent race conditions when assigning unique discount codes concurrently, use Apex record locking (FOR UPDATE) before updating the code status to "Claimed". Flow cannot lock records natively.

2. System Data, Logging & Tracking

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
`_Sent, _Open, _Click`	**Data Cloud:** Engagement DMOs	Legacy System Data Views do not exist on Core. Engagement data streams directly into Data Cloud. Query this data using Data Explorer or the Segment Builder, not standard SOQL.
Custom Send Logging DE	**Core:** Custom Object `Log__c`	If strict audit trails are required, use an Autolaunched Flow triggered by your Campaign Flow to insert records into a custom CRM logging object. Note: Monitor your data storage limits.

3. Advanced Automation & Orchestration

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
SQL Query Studio	**Data Cloud:** Batch Calculated Insights	Standard T-SQL is deprecated. Use Batch Calculated Insights (BCI) in Data Cloud to perform complex aggregations, metrics (e.g., LTV), and multi-object joins at scale without hitting CRM limits.
Script Activity (SSJS)	**Flow:** Scheduled-Trigger Flow	Automation Studio Script Activities must be rewritten as Scheduled Flows. If the logic requires complex looping over thousands of records, use Schedulable/Batchable Apex.

4. APIs & Payload Management

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
`HTTP.Post()`	**Flow:** HTTP Callout Action	Hardcoding API tokens in scripts is obsolete and insecure. Core enforces the use of *Named Credentials* and *External Credentials* in Setup to manage auth securely before the Flow can execute the callout.
`Platform.Function.ParseJSON()`	**Apex:** `JSON.deserializeUntyped()`	Do not attempt to parse 100-node JSON arrays at the exact moment of email send. Shift JSON parsing to the Flow orchestration level or an Invocable Apex method, extract the required strings, and save them to fields.

5. Content & Personalization Context

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
`%%_subscriberkey%%`	**Merge Field:** `{!Record.Id}`	Personalization context is inherited automatically based on the triggering object (Lead, Contact, PersonAccount). External arbitrary keys will break unified tracking.
`ContentBlockByKey()`	**CMS:** Salesforce CMS Workspaces	Legacy Content Builder AMPScript logic is replaced by drag-and-drop dynamic components mapped directly to Salesforce CMS data.

6. Landing Pages & Custom UIs

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
CloudPages & SmartCapture	**Experience Cloud / Screen Flows**	CloudPages processing raw SSJS forms is deprecated. Build public-facing sites using Experience Cloud. Use native Salesforce Screen Flows embedded in the page to handle secure data intake without writing HTML form handlers.
`CloudPagesURL()`	**Core:** Encrypted URL Parameters	To securely pass Contact context to a public Experience Cloud page from an email, you must use Apex Crypto to encrypt the ContactId in the URL, and a Screen Flow to decrypt it on page load.

7. Consent & Subscription Management

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
`LogUnsubEvent()`	**Core:** `ContactPointConsent` Object	Do not use custom boolean fields (e.g., `HasOptedOut__c`) for compliance. Core utilizes a strict Consent Data Model. Updates must target the `ContactPointTypeConsent` or `CommSubscriptionConsent` objects to guarantee global suppression.

8. Cryptography, Hashing & Security

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
`SHA256() / MD5()`	**Apex:** `Crypto.generateDigest()`	Often used to hash email addresses for digital ad networks. Must be processed in Apex before passing the variable to the outbound API or template.
`Base64Encode()`	**Apex:** `EncodingUtil.base64Encode()`	Used for basic credential encoding or file attachment handling. Core natively supports converting Blobs to Base64 strings safely.

9. Error Handling & Transaction Control

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
`RaiseError()`	**Flow:** Custom Error Element	Used to abort a transaction (e.g., stopping an email send if critical related data is missing). The Custom Error element halts the Flow and prevents the DML action.
`try { } catch(e) { }`	**Apex:** `Database.setSavepoint()`	If a multi-step automation fails halfway, MCE left dirty data. On Core, use Apex Savepoints and `Database.rollback()` inside your catch blocks to revert all changes if a single step fails.

10. Array Manipulation & Collections

LEGACY MCE (SSJS/AMP)	SALESFORCE CORE (FLOW/APEX)	ARCHITECTURAL PROBLEM SOLVING
SSJS Arrays `[]`	**Apex:** `List<T> / Set<T>`	Apex is strongly typed. You cannot mix data types in a List. Use `Set<Id>` to automatically deduplicate record IDs before querying to prevent SOQL limit exceptions.
`for (var i=0; i<arr.length; i++)`	**Flow:** Loop Element + Collection Filter	Do not put DML (Create/Update) inside a Loop in Flow. Add modified records to a new Collection Variable, and perform a single Update action on the Collection *after* the loop closes.

Stop rewriting bad code for a new platform.

Migrating to Marketing Cloud on Core is your organization's one chance to eliminate years of technical debt. Do not blindly rebuild legacy scripts 1:1 in Apex.

Book a Codebase Migration Audit