A CIVICSHIFT HANDBOOK

One System, Not Many

A Leader's Handbook for Municipal Data and Information Management

A build-order recipe for organizations starting fresh

Troy Palmer, Founder — CivicShift AI Business Solutions Inc.

Collect once, use many times.

CivicShift AI™

One System, Not Many

A Leader's Handbook for Municipal Data and Information Management

Abstract

Every municipality runs on the same handful of core facts — who works here, what we have custody of, what we spend, what training people have, what incidents occurred. The single most consequential choice an organization makes about information, usually without realizing it, is whether each of those facts will exist once or many times. This handbook is a build-order recipe for getting to once. Written for municipal leaders rather than database administrators, it makes the case for collecting information once and using it many times, then walks through eight steps in the order that prevents the expensive failures: governance before technology, standards before solutions, reference data before transactional records, and privacy designed in rather than bolted on. It closes with the roles that keep the model alive, the anti-patterns that quietly undo it, and checklists a team can put to work on day one. The principles apply to any organization starting fresh — and to any organization willing to start over deliberately.

First edition, 2026.

CivicShift AI Business Solutions Inc. · Fort McMurray, Alberta, Canada

civicshiftai.com

Contents

Contents.....3
How to use this handbook.....4
Part I — The case.....5
Chapter 1. The principle: collect once, use many times.....5
Chapter 2. What happens without design.....7
Part II — The recipe.....9
Step 1. Establish governance before technology.....9
Step 2. Set the standards.....9
Step 3. Define the enterprise information assets.....10
Step 4. Tier the platforms.....10
Step 5. Stand up solution intake.....12
Step 6. Build the reporting layer.....12
Step 7. Design privacy and access in.....13
Step 8. Prepare for intelligence and automation.....13
Part III — Sustaining the model.....15
Roles and stewardship.....15
Anti-patterns: what to watch for.....15
Maturity checkpoints.....16
Appendix A — Starter standards checklist.....17
Appendix B — First ninety days.....18
Appendix C — Glossary.....19

About the Author.....21
About CivicShift.....22

How to use this handbook

This handbook is written for municipal leaders — chief administrative officers, directors, and the managers they trust to build things — not for database administrators. It assumes no technical background. It assumes only that you are responsible for an organization that collects information and reports on it, is accountable for it under law, and would like to do all three well.

It is deliberately framed as a recipe rather than a framework. Frameworks describe an end state; recipes give you an order of operations. The order matters enormously in information management, because most of the expensive failures in this field are sequencing failures: technology purchased before standards exist, solutions built before ownership is assigned, reporting demanded before there is anything authoritative to report from. Follow the steps in Part II in order. Each one is cheap while the organization is small and painful to retrofit once it is not.

The handbook is written for an organization starting fresh — a new municipality, a new division, or a new corporate function standing up its information environment for the first time. If your organization already has years of accumulated systems, the recipe still applies; you will simply be executing it alongside an inventory and remediation effort rather than on a blank page. The principles do not change.

> **Part I** makes the case — the two pictures worth keeping in your head.
>
> **Part II** is the recipe: eight steps in build order.
>
> **Part III** covers how to keep the model alive after the launch energy fades.
>
> **The appendices** contain checklists you can hand to a team on day one.

Part I — The case

Chapter 1. The principle: collect once, use many times

Every municipality runs on the same handful of core facts: who works here, what we have custody of, where things are, what we spend, what training people have, what incidents occurred. These facts are needed by nearly every department, nearly every report, and nearly every decision. The single most consequential choice an organization makes about information — usually without realizing it is making a choice — is whether each of those facts will exist once, or many times.

The default, in the absence of design, is many times. Each department that needs a list of employees builds one. Each program that tracks training builds a tracker. Each team that manages equipment starts a spreadsheet. Every one of these is a rational local decision, made by capable people solving a real problem in front of them. Collectively, they produce the pattern below.

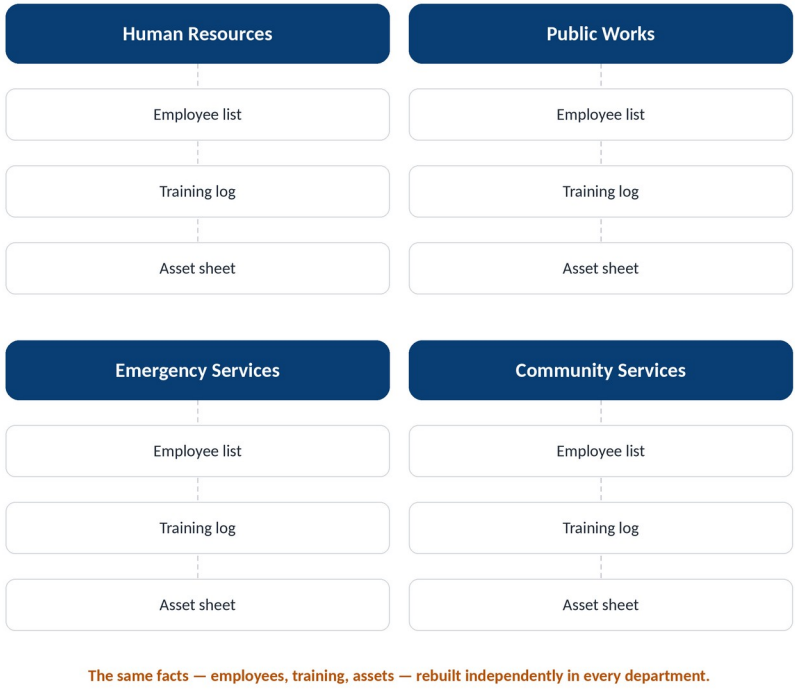

Figure 1. The fragmented pattern — the same data rebuilt in every department.

The cost of this pattern is paid three times. First in entry: the same fact is typed into multiple systems. Second in maintenance: when the fact changes, someone must remember every place it lives. Third in reconciliation: when leadership asks a simple question — how many employees completed mandatory training this year? — three defensible, different answers come back, and someone must spend days determining which is right. Often, none is.

The alternative is not centralizing control of departments. It is centralizing the facts while leaving the work where it belongs. Departments continue to run their own programs, use their own tools, and serve their own residents — but the core facts those activities generate are written to, and read from, one authoritative source per subject.

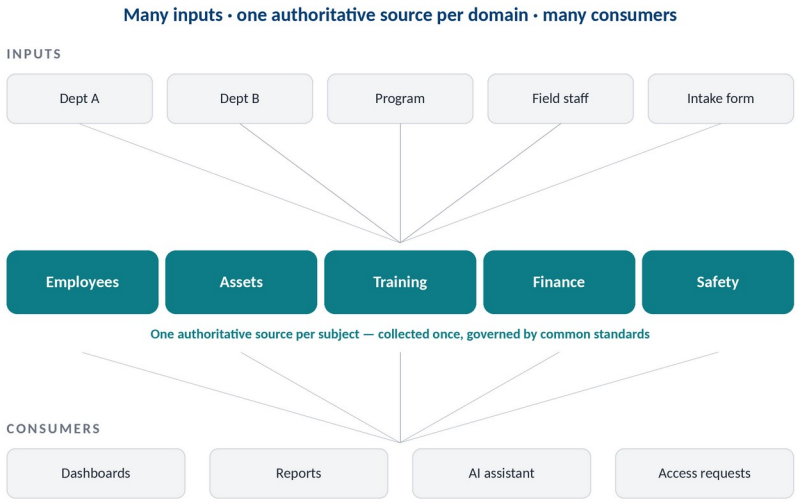

Figure 2. The target model — many inputs, one authoritative source per domain, many consumers.

Read the middle layer of Figure 2 carefully, because it is the entire model. Each teal box is a subject the organization cares about — employees, assets, training, finance, and their peers — existing exactly once, governed by common standards, and consumed by everything below. A department running its own training program does not lose that program; it simply records attendance into the one training source rather than into its own

spreadsheet. Upstream autonomy is preserved. Downstream chaos is eliminated.

> **Many inputs, one authoritative source per domain, many consumers. Collect once, use many times.**

Chapter 2. What happens without design

An organization that skips the design work does not experience a dramatic failure. It experiences a slow accumulation of small ones, each individually tolerable, that compound into five strategic problems.

Decisions lose their footing. When departments define and collect the same measures differently, leadership receives conflicting numbers for the same question. Confidence erodes not because anyone is wrong, but because no source is authoritative. Over time, decision-makers stop trusting reports and revert to anecdote — the most expensive information system ever devised.

Effort is duplicated invisibly. No budget line captures the hours spent re-entering, cross-checking, and reconciling the same information across systems. The cost is real but distributed so thinly across so many people that it never surfaces as a decision. Information should be collected once and reused many times; every additional copy is purely overhead plus error risk.

Knowledge walks out the door. Departmental solutions built without standards depend on the individuals who built them. The spreadsheet only its author understands is a continuity risk wearing a productivity costume. Retirements and departures convert working processes into archaeology.

Security becomes unknowable. Inconsistent site structures, permission models, and sharing practices mean nobody can answer the question 'who has access to what?' — which means nobody can answer 'is that access appropriate?' An enterprise platform without enterprise governance grants enterprise-scale exposure with department-scale oversight.

Legal obligations become unmeetable. Municipalities are public bodies operating under access-to-information and privacy legislation. Responding to an access request within statutory timelines requires locating every responsive record. Privacy legislation increasingly requires knowing where personal information resides, limiting how long it is retained, and reporting

breaches — obligations that presume an inventory. Fragmentation makes each of these a manhunt. Chapter and verse vary by jurisdiction (in Alberta, for example, the Access to Information Act and Protection of Privacy Act), but the pattern is universal: compliance at enterprise scale cannot be achieved through after-the-fact searches. It must be designed in.

And a sixth, about the future.

Artificial intelligence tools — assistants, copilots, automated workflows — depend entirely on structured, governed, trustworthy information. AI does not fix a poor information environment; it amplifies whatever environment exists. An organization with fragmented, inconsistent, ownerless data that deploys AI on top of it will get fragmented, inconsistent, ownerless answers — delivered faster and with more confidence than ever before. The organizations that benefit from AI will be the ones that did the unglamorous work in this handbook first.

Part II — The recipe

Eight steps, in build order. The early steps cost almost nothing but attention; skipping them to get to the visible steps faster is the root of nearly every failure this handbook exists to prevent. A new organization can complete Steps 1 through 3 in its first ninety days with no technology purchases at all.

Step 1. Establish governance before technology

Before any platform is configured, any site created, or any list built, the organization needs three things on paper: a decision-making body, a set of principles, and named ownership. The body — call it an information governance committee — should include business leaders, information management, information technology, and the officials responsible for records and privacy. Its purpose is not bureaucracy; it is ensuring that every future solution lands inside a coherent whole rather than beside it.

The principles fit on one page and should be adopted formally: information is a corporate asset, not a departmental possession; every dataset has exactly one named owner; core facts are collected once and reused; solutions are built to standards; privacy and access obligations are design requirements, not afterthoughts. A one-page charter signed by the chief administrator does more for the next decade of information management than any software purchase.

Ownership deserves emphasis because it is the piece organizations most often skip. Every enterprise dataset in Step 3 gets a named business owner — a person, not a committee, not a department. The owner decides what the data means, who may see it, and how quality is maintained. Where there is no owner, there is no accountability; where there is no accountability, quality falls to whatever level no one has to answer for.

Step 2. Set the standards

Standards are the grammar of the information environment — invisible when present, crippling when absent. A new organization needs six, all writable in weeks: a naming convention for sites, lists, and columns; a metadata standard defining the handful of properties every record carries (owner, classification, retention trigger); an information classification

scheme with three or four levels from public to restricted; a security and permissions model stating how access is granted, by role rather than by individual; a data ownership register; and reporting definitions for the measures leadership will see, so that 'headcount' and 'incident' mean one thing everywhere.

Two disciplines make standards stick. First, identifiers are permanent: once a department code, asset number, or metric key is issued, it is never reused or renamed — records are retired, never repurposed. Second, standards are enforced at creation time through the intake process in Step 5, not audited into existence afterward. Retroactive cleanup is the most expensive activity in information management; the standards exist so it is never required.

Step 3. Define the enterprise information assets

Now the organization decides which facts exist exactly once. The starter list for a municipality is short and predictable: employees and positions; departments and the organizational structure; physical assets; facilities and work locations; training records; safety records; vendors and contracts; projects; and the financial chart of accounts. These are the teal boxes in Figure 2 — the authoritative sources everything else references.

> **Two rules that save years:**
>
> **Build reference data before transactional data.** The list of departments comes before the incident register that assigns incidents to departments; the employee source comes before the training records that point at employees. Every transactional record built before its reference data exists will carry free-typed values that someone must later untangle.
>
> **Prefer records over totals.** Capture one row per real-world event with its attributes, not monthly summary counts. Counts can always be computed from records; records can never be recovered from counts. 'A grievance filed on this date, by this group, in this department, resolved in this many days' answers questions you have not thought of yet. 'Four grievances in March' answers exactly one.

Step 4. Tier the platforms

Modern productivity suites blur the line between document storage, lightweight lists, and databases — and the blurring is where organizations get

hurt. The tiering principle: match each dataset to the platform tier its volume, criticality, and integrity requirements demand. Document libraries hold content — files, correspondence, records of decisions. Lightweight lists suit small, single-team, low-volume structured data. Enterprise information assets — the Step 3 list — belong on a relational database platform.

The distinction is structural, not cosmetic. In a flat list, a column like 'Department' is just text; nothing stops one record from saying PW, another Public Works, and a third P.W. In a relational platform, that column is an enforced reference to the one department table — the platform refuses anything else. From that single property flows most of what enterprise data management requires: rename a department once and every record reflects it; deletions can be blocked while dependencies exist; security can be applied to the data itself, row by row and column by column; and every change is captured in a built-in audit trail — the difference between asserting compliance and demonstrating it.

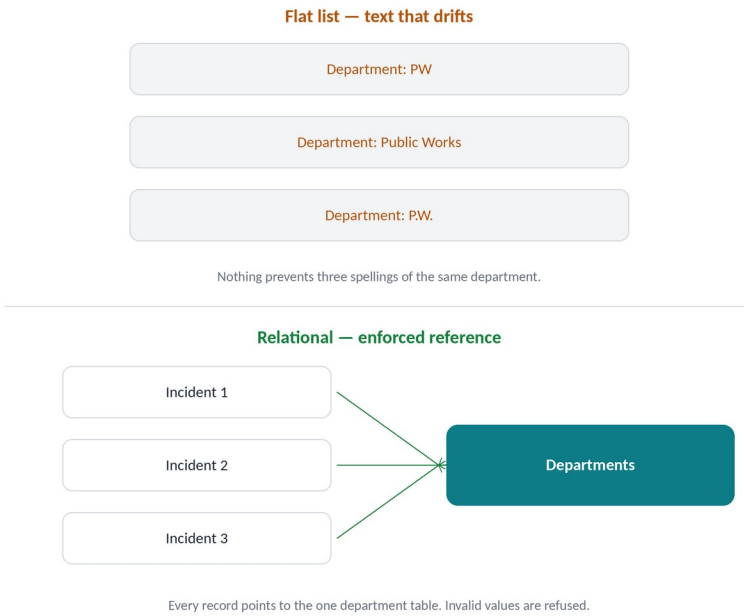

Figure 3. Why enterprise data belongs on a relational platform — enforced references versus drifting text.

Flat lists also carry hard technical ceilings — item thresholds and query limits that degrade performance as data grows and, in some configurations, silently return incomplete results. A list that works beautifully at five hundred rows becomes a liability at fifty thousand. The tiering decision made at creation time costs nothing; the migration forced later costs a project.

Licensing cost is the standard objection to the relational tier, and the tiering principle is the answer: nobody proposes putting every dataset on a database. The relational tier is for the small number of high-value enterprise assets where integrity, security, and audit are requirements. Everything else stays on the lighter, cheaper tiers where it belongs.

Step 5. Stand up solution intake

Departments will build things — that energy is an asset, not a threat. The organization's job is to channel it through a lightweight front door. Before a new site, list, application, or automated workflow is created, a short intake asks five questions: What business problem does this solve? What data will it hold, and does any of it already exist as an enterprise asset it should reference instead of duplicate? Who owns it? What classification applies? What is its expected volume and lifespan?

Keep the process fast — days, not weeks — or it will be routed around, and shadow systems will bloom in exactly the dark corners governance was meant to illuminate. The intake exists to make good solutions land inside the model, not to prevent solutions. It also produces something quietly invaluable as a by-product: a living inventory of every system, dataset, and report in the organization, which is the first thing an auditor, a privacy commissioner, or an incoming leader will ask for.

Step 6. Build the reporting layer

Reporting is where the model pays for itself visibly. Because the enterprise assets are authoritative and consistently structured, reports draw from one source and agree with each other by construction. The recipe within the recipe: define each measure once, in writing, with its owner (what counts as an 'active recruitment'? as of what moment in the month?); compute measures from the record-level data rather than asking departments to submit totals; automate the flow from source to dashboard so refresh is

scheduled, not manual; and route any remaining manual entry through structured forms that validate against reference data at the point of entry.

Design for the trajectory from manual to automatic. Early on, some measures will be typed in monthly by program staff; over time, more flow directly from source systems. A well-designed reporting model makes that transition invisible — the measure's definition and history are unchanged; only its entry method evolves. Reserve summary submission for the cases where record-level capture is genuinely impractical, and treat each one as temporary.

Step 7. Design privacy and access in

Public bodies do not get to treat privacy and access as compliance paperwork bolted on at the end; the obligations are architectural. The good news is that an organization following Steps 1 through 6 has already done most of the work: classification exists (Step 2), ownership exists (Step 1), personal information lives in known, governed locations (Steps 3 and 4), and an inventory exists (Step 5). What remains is to connect that architecture to the statutory machinery.

Concretely: maintain a personal information inventory — which assets contain personal information, why it is collected, and its retention period; apply retention and disposition schedules through the metadata standard so records age out by rule rather than by memory; ensure access requests can be executed as searches across governed sources rather than expeditions across shared drives; and establish breach response procedures that can answer, quickly, exactly what information a compromised location contained. Privacy impact assessment belongs in the Step 5 intake for any solution touching personal information — cheap when done at design time, humiliating when done after deployment.

> **Over-retention is the failure mode nobody sees coming.** Keeping personal information forever feels safe and is the opposite: most modern privacy statutes treat retaining personal information beyond its purpose as a violation in itself, and every forgotten repository is breach surface. Disposal, on schedule, by design, is a compliance activity.

Step 8. Prepare for intelligence and automation

The final step is mostly a payoff for the previous seven. AI assistants, automated workflows, and advanced analytics all consume the same fuel: structured, classified, owned, trustworthy information. An organization arriving at this step with the model intact can adopt these tools quickly and safely — the assistant answers from authoritative sources, the automation writes to governed registers, and the security trimming that protects sensitive information in the hands of staff protects it identically in the hands of the machine.

Two governance additions complete the picture: extend the Step 5 intake to cover AI use cases (what data will the tool read? is any of it personal or restricted? who is accountable for its outputs?), and treat AI-generated content as records like any other — classified, retained, and disposable under the same schedules. The organizations that struggle with AI governance are, almost without exception, the ones that never established information governance; the former is not solvable without the latter.

Part III — Sustaining the model

Roles and stewardship

The model survives on four roles. The executive sponsor — ideally the chief administrator — holds the principle that information is a corporate asset and settles the disputes that principle creates. Data owners, named in Step 1, are accountable for the meaning, quality, and access rules of their assets. Data stewards do the daily work owners are accountable for — maintaining reference data, reviewing quality, handling access requests within their domain. Information technology provides and secures the platforms, but pointedly does not own the data; the moment data ownership migrates to IT, business accountability evaporates and quality follows it.

Stewardship is a role, not usually a job. In a small municipality, the same person may steward three assets alongside their day job; what matters is that the name is written down and the duties are explicit. Budget a small amount of recognized time for it. Stewardship that exists only as an unfunded expectation is the first thing dropped in a busy month, and data quality decays at precisely the speed of dropped stewardship.

Anti-patterns: what to watch for

Certain failure patterns recur so reliably across organizations that they deserve names. Watch for these; each is cheap to correct early and brutal to correct late.

The convenience copy. Someone exports the employee list to a spreadsheet 'just for this analysis.' Six months later the copy is being maintained in parallel, has drifted from the source, and feeds a report. Every copy of enterprise data outside its authoritative home should have an expiry date and a delete habit.

The renamed identifier. A code or key is reused for a new purpose because creating a new one felt like clutter. Every historical record carrying the old meaning is silently corrupted. Identifiers are permanent; retire, never recycle.

The hero system. One capable person builds an impressive solution nobody else understands, and the organization quietly becomes dependent

on their continued employment and goodwill. Admire the initiative; require the documentation and the standards compliance.

The totals trap. A new tracking need is met with a monthly-summary spreadsheet because it seems simpler than a proper register. Within a year, leadership asks a breakdown question the totals cannot answer, and the history is unrecoverable. Capture records; compute totals.

The permission snowflake. Access granted person-by-person, exception-by-exception, until the permission model is an archaeology site nobody dares touch. Grant by role, review on a schedule, and treat every individual exception as debt.

The governance theatre. A committee that meets, minutes, and approves, while solutions are built around it because its process is slow. Governance earns its authority by being fast and useful; the moment it is neither, it is decoration.

Maturity checkpoints

Progress in information management is hard to see day to day, so measure it annually against questions rather than scores. Can any leader get the authoritative answer to a core question — headcount, asset count, training compliance — from one place, without asking a person to compile it? Does every enterprise dataset have a named owner who knows they own it? Can the organization produce, within a day, an inventory of systems holding personal information? When a department proposes a new solution, does it come through intake as a matter of habit? When someone leaves, do their systems keep working?

> *The end state worth aiming at is quiet: reports that agree with each other, access requests that are searches rather than expeditions, new solutions that land inside the model without ceremony, and an AI assistant that can be trusted because the information underneath it can be. None of it is glamorous. All of it compounds.*

Appendix A — Starter standards checklist

The six standards from Step 2, expanded into a working checklist. Target: all six adopted within the first ninety days.

- ☐ **Naming convention adopted**— sites, lists, tables, and columns follow a published pattern; abbreviations come from a maintained list.
- ☐ **Metadata standard adopted**— every record type carries owner, classification, and a retention trigger at minimum.
- ☐ **Classification scheme adopted**— three or four levels (e.g., public, internal, confidential, restricted), each with handling rules.
- ☐ **Security model adopted**— access granted by role; individual exceptions logged and time-limited; access reviews scheduled.
- ☐ **Data ownership register created**— every enterprise asset lists a named owner and steward; register reviewed annually.
- ☐ **Reporting definitions written**— each leadership measure has a written definition, owner, and calculation method.
- ☐ **Identifier policy adopted**— codes and keys are permanent; retirement, not reuse; no meaning embedded in identifiers that can change.
- ☐ **Retention schedule adopted or inherited**— every record class maps to a retention period and disposition action.

Appendix B — First ninety days

A build-order checklist for a new organization, drawn from Part II. Technology purchases appear only in the final third.

Week 1–2: Executive sponsor named. One-page information principles charter drafted and signed.

Week 2–4: Governance committee constituted — business, IM, IT, records, privacy. Meeting rhythm set (monthly, one hour).

Week 3–6: Six starter standards drafted (Appendix A). Adopted by the committee, published where builders will find them.

Week 4–8: Enterprise information assets listed. Owners and stewards named for each. Reference data built first — departments, locations, position types — before any transactional register.

Week 6–10: Platform tiers decided — what lives in document libraries, what in lightweight lists, what on the relational tier. Licensing sized to the enterprise-asset list only.

Week 8–12: Solution intake process published — five questions, answered in days. Inventory of approved solutions begins as its by-product.

Week 10–13: First enterprise assets stood up on the relational tier. First reporting measures defined in writing and connected to record-level sources.

Ongoing: Personal information inventory maintained. Privacy impact assessment wired into intake. Annual maturity checkpoint scheduled.

Appendix C — Glossary

Authoritative source — The single, governed location where a category of information officially lives. All other appearances of that information are references to it or copies with expiry dates.

Classification — A label describing how sensitive information is and therefore how it must be handled, stored, shared, and disposed of.

Data owner — The named businessperson accountable for a dataset's meaning, quality, and access rules. Distinct from the IT staff who operate the platform it sits on.

Data steward — The person who performs the day-to-day care of a dataset on the owner's behalf — maintaining reference values, reviewing quality, actioning access requests.

Disposition — The scheduled end-of-life action for a record: destruction, anonymization, or transfer to archives.

Enterprise information asset — A dataset important enough to the whole organization that it is managed as a single authoritative source — employees, assets, training records, and their peers.

Intake — The lightweight front-door process a proposed new solution passes through, ensuring it lands inside the governed model rather than beside it.

Metadata — Structured information about a record — its owner, classification, dates, retention trigger — that makes it findable, governable, and disposable by rule.

Personal information inventory — The maintained list of which systems hold personal information, why it was collected, and how long it is kept. The foundation of privacy compliance and breach response.

Reference data — The stable lookup lists — departments, locations, categories — that transactional records point to. Built first, changed rarely, owned explicitly.

Relational platform — A database platform in which relationships between tables are enforced by the system itself: a record cannot reference a department that does not exist. Contrast with flat lists, where such references are unvalidated text.

Retention schedule — The rulebook mapping each class of record to how long it must be kept and what happens to it afterward.

Shadow system — A dataset or application created outside the governed model — usually a spreadsheet — that the organization depends on without knowing it exists.

Transactional data — Records of events and activities — incidents, requests, hires, purchases — as distinct from the reference data they point to.

About the Author

Troy Palmer is the creator of the CivicShift Operating Model™, a Canadian-registered organizational framework designed to help multi-service, multi-department organizations operate with greater clarity, consistency, safety, and accountability.

Drawing from more than 25 years of public service — including 18 years as a professional firefighter and EMT — Troy has built a career at the intersection of leadership, operations, safety, and digital modernization. He has held leadership roles in emergency services, human resources, safety, municipal program management, organizational development, and digital transformation. Across each of these domains, he witnessed the same recurring patterns inside complex organizations: fragmented data, unclear roles, process drift, unmanaged risk, and technology implementations that failed to translate into meaningful operational improvement. Those lived experiences shaped a framework that unifies leadership systems, performance systems, governance, safety integration, data quality, and AI-enabled decision support into one coherent structure.

"Organizations succeed when people have the information they need, when they need it, and before they need it."

Troy holds a Bachelor of Applied Business degree in Emergency Services Management, earned with distinction, along with an Emergency Services Technology diploma and MIT Sloan executive education on the implications of AI for business strategy. His Microsoft 365–aligned approach leverages SharePoint Online, Power Apps, Power Automate, Power BI, and modern governance practices to help organizations build structures that are scalable, traceable, and ready for the future.

Beyond the professional record, Troy's deepest sense of purpose comes from raising his daughter. Many years of memories, school routines, athletics, camping, traveling, and everyday life have shaped his understanding of resilience, responsibility, and what it means to show up fully — for family, for community, and for the people depending on him.

Troy lives and works in Fort McMurray, Alberta.

About CivicShift

CivicShift AI Business Solutions Inc. is a municipal technology consultancy based in Fort McMurray, Alberta. Built on Microsoft Dataverse and the Power Platform, CivicShift delivers governance, safety program management, reporting, and ATIA/POPA-compliant document and records management solutions designed specifically for Canadian municipalities and public-sector organizations.

The CivicShift Operating Model™ pairs a human operating system — leadership, roles, culture, and safety — with a machine operating system: catalogs, flows, governance, reliability signals, and explainable AI pathways that give organizations the data backbone to operate at scale. Every solution deploys into the client's own Microsoft 365 tenant, so municipalities keep full ownership of their data and their environment.

To talk about bringing one system — not many — to your organization:

CivicShift AI Business Solutions Inc.

civicshiftai.com

troy.palmer@civicshiftai.com · 780-792-7227

Notes

Collect once, use many times.

“Let computers do the computer stuff”

CivicShift AI™

civicshiftai.com