**NCCSD Systems Workgroup Vendor Forums – Q&A related to “Low Code/COTS”**

Vendor Name: Cambria Solutions, Inc.

Please enter your responses into this document, but feel free to send any other attachments as well.

Questions:

1. Since there is not yet a consistent term or definition for this approach, please give your company's description, including your terminology and definitions. How is this approach different from a "custom" build of a child support system? If you choose to do a quick demo or screen shots that would be welcome.

COTS refers to a commercial-off-the-shelf product (“shrink wrapped”) that is designed to be flexible and work with minimal customization in a customer’s environment. The product is already generally available, and a customer can buy it “off the shelf”. Microsoft Office is an example of such a product.

According to Gartner, Platform as a service (PaaS) (or just, Platform) is application infrastructure functionality enriched with cloud characteristics and offered as a service. Application platform as a service (aPaaS) is a PaaS offering that supports application development, deployment and execution in the cloud. It encapsulates resources such as infrastructure. High-productivity aPaaS (hpaPaaS) provides rapid application development (RAD) features for development, deployment and execution — in the cloud. The rapid application development (RAD) features are often referred to as "low-code" and "no-code" support. These hpaPaaS solutions contrast with those for "high-control" aPaaS, which need professional programming — "pro-code" support, through third-generation languages (3GLs) — and provide transparent access to the underlying infrastructure.

There are different types of low/no-code platforms. There is the business-process management (BPM) type, which would include applications such as Salesforce, Microsoft Dynamics, Pega Systems, MicroPact, and even SAP and PeopleSoft to an extent. The applications are designed for versatility in that they can function as a case management system, a financial system or a workflow system for any industry or any governmental program area. Then, there is the web content-management type: WordPress, Joomla, Dot Net Nuke, SharePoint and a host of others. These are generally geared at rapid site development, published content management and libraries. They are built to make the content management, publishing and workflow manageable by business users. Finally, there is the cloud-based builder apps like Wix, GoDaddy, SiteBuilder and others that drive web functionality through a templated framework for the general public. Their purpose is to make the internet accessible to everyone and anyone that can point, click, or drop paragraphs, graphics, content and pre-built services. A large chunk of what is seen every day on the Internet is driven by this technology.

As stated earlier, developing on a low/no-code platform is vastly different from developing from scratch (or custom or pro-code). With a low/no-code platform, one can expect to see functional systems with some significant benefits:

- **Productivity:** More rapid implementation to delivery functionality to customers over custom coding.
- **Responsiveness:** No-code platforms are not susceptible to hand-coding errors. Visual modeling is intuitive, and UI components and data service integrations make it possible to build working

prototypes in just days of developing. At this speed, it is easy to focus on an output of working software that meets user requirements. Plus, all of the UI is prebuilt to run on multiple platforms from PC to Tablet to Mobile.

- **Reliability:** Maintenance comes from the platform vendor who has pretested new releases, so they are generally more reliable.
- **Time and Cost Savings:** Developers can generate more functionality in a shorter period of time, adding business value in needing fewer coders to achieve the same result. Over-time, maintenance of the application and total cost of ownership (TCO) is lower as changes can be done by configuration and less customization.
- **Innovation in the hands of people, not systems:** Developing with no-code/low-code sees people as the core over processes and tools. By removing the hurdle of programming, no-code opens the developer playing field to a much wider audience than the limited population of highly specialized programmers. Customers become active participants and they can contribute their own knowledge. This ensures that the solution achieves what it set out to. Incorporating an element of "DIY" to application development means happier customers and time saved by eliminating revisions.

2. With reference to the "core" functionality required by the OCSE Systems Certification Guide (Case Initiation, Locate, Establishment, Case Management, Enforcement, and Financial Management), how does this approach handle each area? In particular, since Child Support requires complicated financial processing, e.g. distribution rules and arrears calculations, please address how these are handled with this approach.

With a Platform approach, the "core" functionality modules are developed and implemented using the rapid application development methods supported by the platform. In order to do this, one does not have the start from scratch with the barebones platform. The platform provides for common services, such as data access, security, reporting, auditing, workflow, correspondence, and rules processing. In addition, most platform vendors have vertical solutions and accelerators that allow reuse of proven functionality to be the basis for a head start. This way, a platform can hit the ground running and the customers are able to look past the basic (core) functionality, such as case management and financial management, and expend valuable resources on developing customizations, integrations, and what is not available out-of-the-box.

In the case of Microsoft Dynamics, there are many functional areas or modules available and financials, in particular, are included as part of Finance and Operations module, which is the ERP component. Financials is often a missing component with some platform products that tend to focus mostly on the CRM side of things, but with Dynamics it is part of a mature ecosystem of integrated products and functionality.

And although not a preferred approach, a platform can also facilitate a way to host 3rd party modules. This allows for hybrid solutions, where part of the functionality is developed with the inherent development capabilities of the platform and the rest is implemented via commissioning of 3rd party modules. In these situations, the platform provides a way for the modules to communicate with one another. The platform facilitates integration – the modules can talk to the platform in any language and the platform is responsible for translating a request from one module to another in a form that the receiving module can understand and respond to. This also allows for plug-and-play of modules – one could get best of breed modules and have them interoperate.

One example of a 3rd party integration for Child Support would be a robust business rules management system (BRMS). While Dynamics and other Platform products do include business rules capabilities out-of-the-box, they are designed more for forms logic and recommendations. But highly complicated rules logic and calculation processing based on complex data may require a more robust and specialized product. The good news is there are BRMS products, such as InRule, that are designed to integrate easily with Dynamics through pre-built configurations and avoid the need to build custom solutions that are costly and hard to maintain. A specialized enterprise BRMS would enable complex business logic to be handled, such as financial processing and calculations, and provide an intuitive and proven product to write, manage, share and execute decision logic. InRule or other 3rd party BRMS solutions are commonly used in industry solutions that require similar financial complexity including healthcare, insurance, and financial services.

3. What COTS or other products are used in conjunction with this approach to give a state a fully functional system?

We would use a high-productivity aPaaS (hpaPaaS) platform that provides rapid application development (RAD) features for development, deployment and execution — in the cloud. The platform would form the low-code/no-code foundation that will provide all the benefits highlighted in #1 above. Cambria would propose to use Microsoft Dynamics as the platform of choice, which would also include a full financial ERP solution.

In addition, and based on the state-specific requirements, there may be infrastructural, testing, and/or integration tools needed. Preference will be given to tools that have been proven to work with Dynamics. Such tools would include, but not be limited to, document management, data warehouse and analytics, e-signature, content management, single sign-on, smart search, and integration with channels such as voice and mobile. Most of these tools are already available in the Microsoft Azure Marketplace and can be integrated and deployed easily with Microsoft Dynamics. Microsoft Azure DevOps provides a whole ecosystem of integrated tools and services to support teams to plan work, collaborate on development, and build and deploy applications.

And as previously mentioned in #2 above, a 3rd party business rules management system (BRMS), such as InRule, would likely need to be used in conjunction with Dynamics.

4. Under what circumstances does it make the best sense for a state child support agency to consider this new approach versus other possible means of modernizing its child support system? Are there any characteristics of either a state's IT system or its business processes that lend themselves more to this approach?

Often, two overarching factors – budget and availability of state resources – weigh heavily on the decision to refactor/re-platform an existing system versus design, develop, and implement an entirely new Platform solution. The latter can require a much heavier investment of time and money, with a higher degree of risk. But, the benefits of implementing a new Platform solution can be significant and may be what a state child support agency needs to execute their vision and achieve their business goals and objectives.

In terms of specific IT or business process characteristics, there are several key considerations that often influence a modernization path.

- **Maintenance Cost** – It's no secret that the cost of maintaining legacy systems with physical servers, especially a mainframe, can be significantly higher than current cloud hosting options. The potential cost savings of migrating to a cloud environment, whether re-platforming or refactoring or building a new Platform solution, can go a long way towards offsetting the cost of undertaking a project.
- **Technical Support** – If current systems are difficult to maintain due to unsupported technology stacks and/or skilled resources not available in the market to perform ongoing maintenance and operations, then re-platforming/refactoring may be a good option.
- **Enterprise Approach** – If an agency wants to reuse the platform across multiple programs and adopt an integrated enterprise standard, then a Platform solution can be the way to go and provide economies of scale in terms of licenses, costs, skills, and training required.
- **Functional Fit** – If current systems do not adequately meet business needs and outcomes (e.g., program rules, regulations, policies, process efficiencies, etc.) then a new Platform solution may be a better way to go. Otherwise, just re-platforming something that doesn't functionally work well, may still require a high degree of refactoring and modernizing in order to make necessary changes to move the needle on achieving better business outcomes.
- **Requirement Changes** – If program rules, regulations, policies, business requirements or processes are frequently evolving or changing, then a new Platform solution may be able to better handle such changes. As described previously, a low-code/no-code platform should be able to implement changes rapidly and without the need for custom coding and extensive testing that must go with it.
- **Standard Processes** – The maturity curve on Platform solutions has been rapidly increasing over the last several years. Industry solutions that are repeatable are constantly being designed and configured on low-code/no-code platform offerings. If standard business processes can leverage a pre-configured Platform solution as a baseline starting point, rather than starting from scratch, then it can rapidly accelerate a project and reduce the overall cost and risk of implementing a new Platform solution.
- **Mobility and Offline Access** – If the need for access on mobile devices and/or working in a disconnected mode (e.g. offline work in the field) is part of the solution capabilities desired, then modern Platform solutions can often provide that out of the box. Many Platform products have embedded these capabilities in their latest architectures and, therefore, make it easy for customers to enable these features and realize these added benefits with little to no additional cost.

5. Generally speaking, what should a state expect on the following: project timeframe, project cost, time to rollout statewide?

A platform-based approach would allow for a quicker rollout than a conventional, custom-development approach. The project timeframes and costs depend on the state-specific requirements, constraints, size, complexity, integration, and implementation needs. However, generally speaking, the states should expect the rollout to be completed between 24 to 36 months with an initial cost of $10M - $40M.

6. The states don't want to again face the major system build and cost challenges once they have modernized. If they choose this approach, what is the continuous improvement model for the platform? Will the states benefit from the vendor efforts without major costs?

A platform-based approach is ideal to take advantage of continuous improvement provided by the platform vendor. The functionality is built on top of the technology foundation provided by the platform. This provides a level of abstraction between the technology and the functionality layers, where one could enhance both layers at the same time, or at different times. As the technology advances and the platform vendor enhances the platform to take advantage of the new advances, it is easy to upgrade the platform without disturbing the functionality built on top of the platform. Also, the platform typically runs in the cloud and is immediately able to benefit from the upgrades provided by the vendor. As resources permit, the functionality can be enhanced over time to take advantage of the technical advances. Most platform vendors, including Microsoft, provide a roadmap of the future features and releases that will help the states plan as to when to best enhance the functionality. When the platform is enhanced, the cost per user typically remains the same, therefore the states benefit from the newly added features and the vendor's efforts without incurring major costs.

7. What are the most important things that a state should do to prepare for this approach?

If a state decides a new Platform solution is the best approach for them, then properly preparing for such an endeavor is critical and can make a big difference is the success of a project and eventual outcomes achieved.

- **Vision and Goals** – Establish a common enterprise vision and organization goals up front. This should be reflected in a Project Charter and sets the direction and expectations for delivering a successful project.
- **Roadmap** – Create a high-level roadmap/blueprint for how the project will be done. This will continue to evolve as the project gets underway and progresses, but it provides necessary guidance and milestones for how the project is envisioned to be completed.
- **As-Is Analysis** – Before building something new, it is best to understand what you already have – good or bad. Gather or create documentation, to the extent possible, on the following areas:
  - Business processes
  - Legacy systems
  - Data models and dictionaries
  - Business rules
  - Policies and procedures
  - Pain points/opportunities for improvement
  - User roles/personas
- **To-Be Inventory** – Analyze and define the business inventory of what needs to be included in a new Platform solution. Essentially this starts to create the backlog of work or scope items that must be included.
  - Application Features/Functionality
  - Business Processes/Use Cases
  - Business Rules
  - Dashboards/Reports/Data Analytics
  - System Interfaces

    - Data Migration
    - Workflows
- **User Experience** – Begin thinking about the desired user experience. Often systems are built to meet top-down program policies and requirements but fail to provide a well thought out experience for its users/citizens. Defining user personas and journey maps can provide valuable groundwork for a future Platform consideration when it comes to modern user experience and user interactions.
- **Project Resources** – Understand the reality of resource availability and skills within your organization. By understanding this upfront, you can avoid the pitfalls of selecting a project delivery approach that does not align with the staff or skills available to you.
- **Change Management** – Assess current organizational desire and ability to embrace and manage change. System projects can bring about the hope of many long-awaited new benefits and improvements, but often not without a high degree of change. This requires change management to be an intentional and integral part of a project from the beginning, and not left to just before go-live, or worse, a reactive after-thought once significant challenges arise.

8. How does this type of child support system fit with states who need to have an enterprise approach? Many of the platforms seem to be creating the same old silos on a new platform. Is it possible to have one casefile for each person/family across the systems (child support, SNAP, TANF, family services, etc.)?

Yes, it is possible to have an enterprise approach with a platform. The enterprise design and development standards can be applied when building with a platform. Platforms lend themselves extremely well to agile development or hybrid development approaches. The users can be very involved with the development process, thus ensuring early testing and buy-in. The functional assets developed can go through the enterprise configuration management and software deployment processes. Further, an enterprise integration and data architecture approach can ensure that the data stored inside a platform is easily available for enterprise use. This can be achieved by real-time or near-real-time replication of data from the platform store to an enterprise store. Therefore, it is entirely possible to have one casefile for each person/family stored in the platform for child support purposes, and then replicated to an enterprise data store for other program purposes.

9. What haven't we asked that we should have?