Efficient Energy Research: Building an Advanced Language Model and Interface

STUDENTS: Najib Haidar, Whitney Waldinger, (Gerald) Ichiro Nakata, Akash Shetty, Brian Han, Aaron Hong, Benjamin Jiang, Joni Nguyen

ACEP
Alaska Center for Energy and Power

Background

Goal: Create an open source LLM that can be utilized to assist in finding energy research documents to streamline answering of researcher's questions.

Requirements:

- LLM able to respond based on database of energy research documents and hosted on a website.
- Open-source resources (as much as possible).
- Modular components for data pipelines.
- Return the documents from which the response is generated.

Our Approach

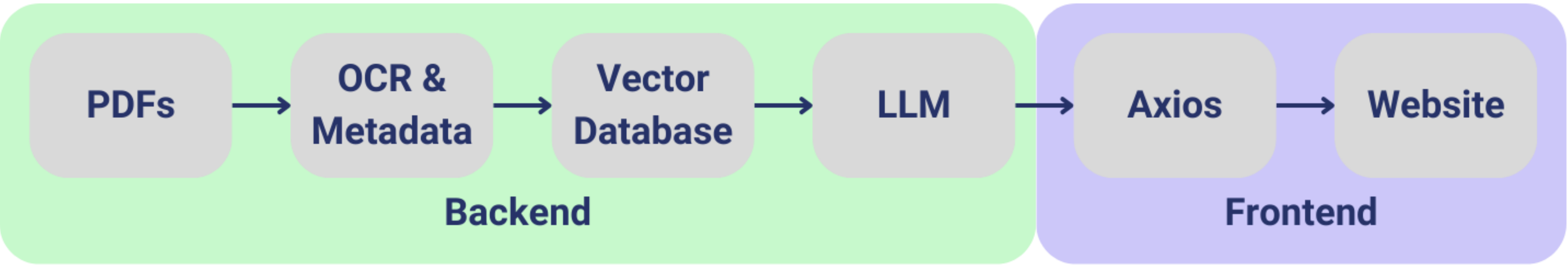


Fig 1. Overall Project Diagram

- Used pre-existing models to streamline production process.
- Data processing pipeline constructed to handle PDF input to embeddings.
- Query response pipeline handles RAG (Retrieval Augmented Generation) functionality using the LLM.
- Website created to allow for interaction with the backend components.

Frontend Connection

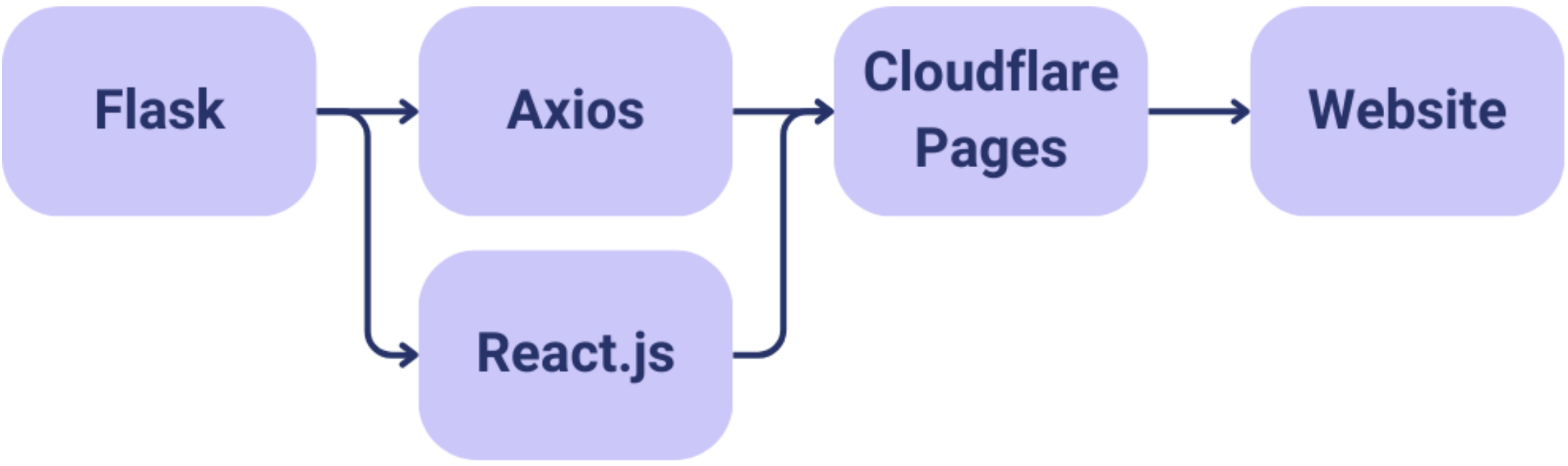


Fig 2. Frontend Flow Chart

- **Flask**: Acts as the backend framework that mediates interactions between the frontend and the LLM.
- **Axios**: HTTP client used in React App to send requests to backend.
- **React.js**: Framework used to build the interface of the website.
- **Cloudflare Pages**: Deployment platform for the website with seamless Git integration for UI updates. Has built in security.
- **Website**: The user access point for interacting with the system.

Data Input Workflow

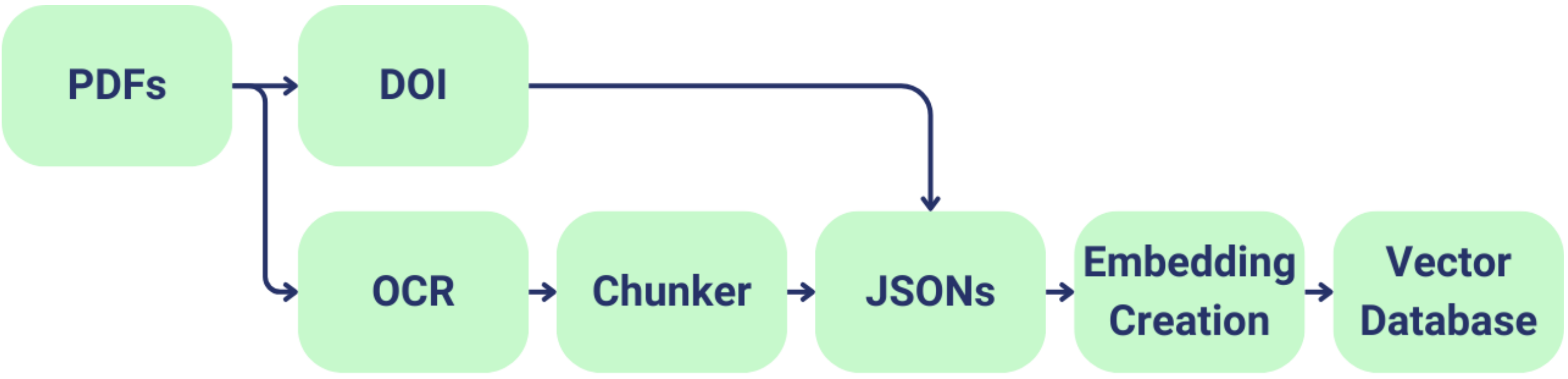


Fig 3. Data Input Flow Chart

- **PDFs**: The data input pipeline accepts documents in PDF form for processing.
- **DOI (Digital Object Identifier)**: A database which is queried to find any metadata associated with the document.
- **OCR (Optical Character Recognition)**: A machine learning model to retrieve the text from the PDF document. The currently used model is Tesseract.
- **Chunker**: Breaks the text into smaller sized files or chunks of text. This is important for LLM prompt token limitations.
- **JSONs**: A composite data file with the textual data from the OCR and Chunker as well as the metadata retrieved from the DOI database.
- **Embedding Creation**: The conversion of textual format to a numerical vectorized format. This is handled by Together AI using the m2-bert-80M-8k-retrieval model.
- **Vector Database**: The storage and reference module for the vectorized embeddings of the JSON files. MongoDB Atlas was utilized.

Query Response Generation

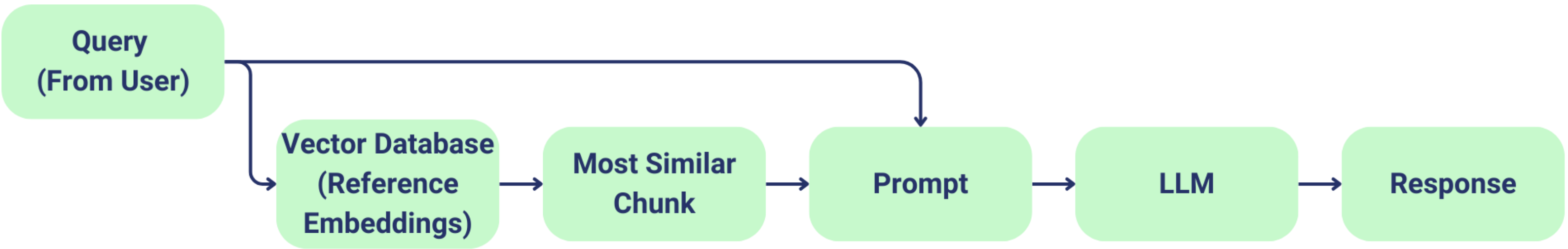
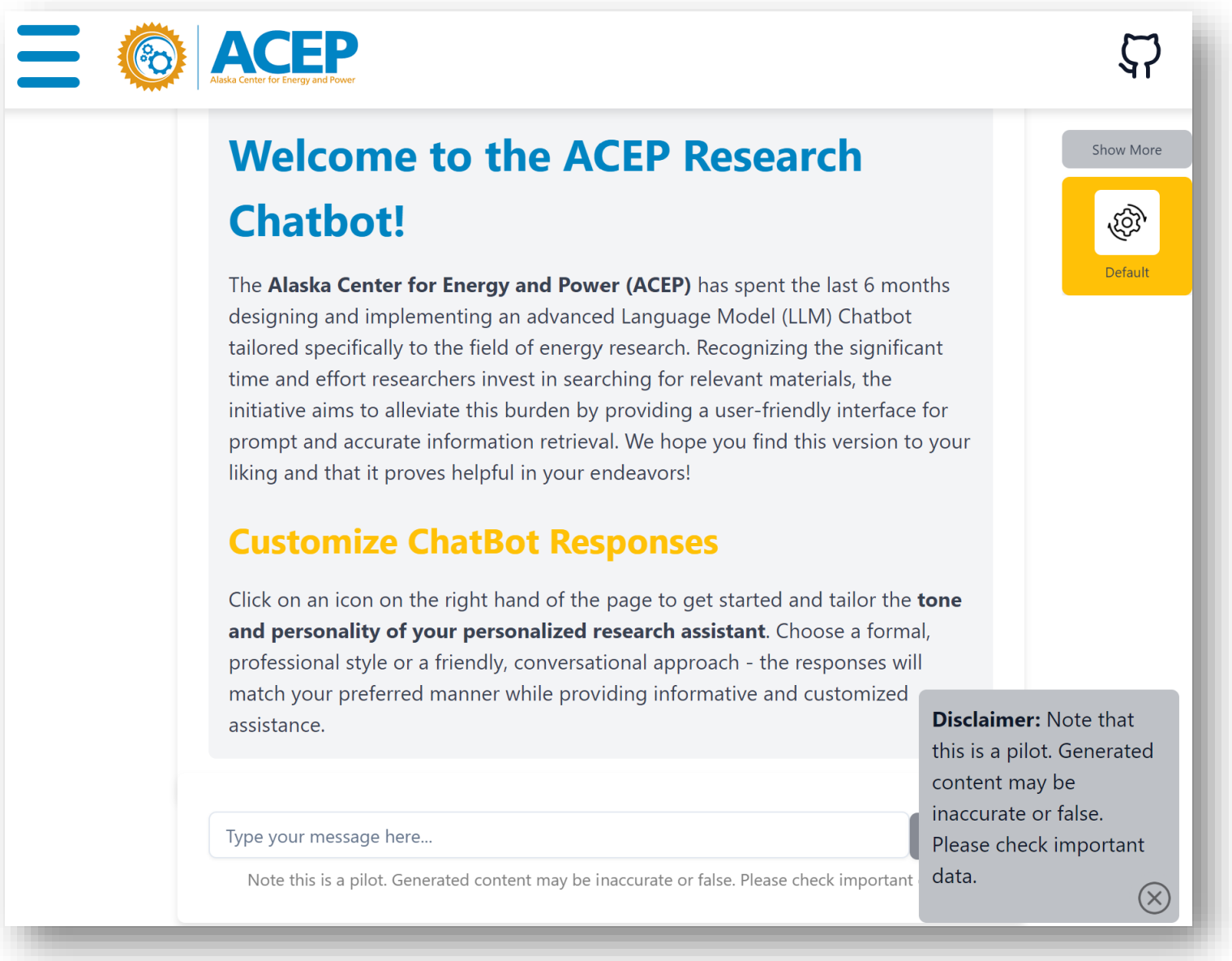


Fig 4. Query Response Flow Chart

- **Query**: The question/prompt sent by the user. In our build sent by Axios.
- **Vector Database**: The embeddings which the query is compared to, so that the most similar embedding can be found.
- **Most Similar Chunk**: The chunk returned by the Vector Database in JSON format.
- **Prompt**: A composite block of information containing the Query, Most Similar Chunk, and statements to tune the LLM generation.
- **LLM (Large Language Model)**: Generates the response with the given information from the prompt. The model used is Llama3.
- **Response**: The reply for the query generated by the LLM using the prompt. This is returned to the user through the website.

Website



Welcome to the ACEP Research Chatbot!

The **Alaska Center for Energy and Power (ACEP)** has spent the last 6 months designing and implementing an advanced Language Model (LLM) Chatbot tailored specifically to the field of energy research. Recognizing the significant time and effort researchers invest in searching for relevant materials, the initiative aims to alleviate this burden by providing a user-friendly interface for prompt and accurate information retrieval. We hope you find this version to your liking and that it proves helpful in your endeavors!

Customize ChatBot Responses

Click on an icon on the right hand of the page to get started and tailor the **tone and personality of your personalized research assistant**. Choose a formal, professional style or a friendly, conversational approach - the responses will match your preferred manner while providing informative and customized assistance.

Type your message here...

Note this is a pilot. Generated content may be inaccurate or false. Please check important

Disclaimer: Note that this is a pilot. Generated content may be inaccurate or false. Please check important data.

Fig 5. Website Interface for ACEP LLM

LLM Chatbot Link

Future Work

There are a few different ways in which the project could be further developed.

- **Formalized Testing**: A formalized method of testing the capabilities of the chatbot's recall and generation capabilities could be developed. The LLM is already an established model, so the testing would cover our application of it.
- **Streaming Response**: LLMs typically take a while to generate and send the full response, so streaming the response in chunks can be done to provide a more seamless response for queries.
- **Saved Chat History**: Saving chat for different users would be useful for picking up where researcher left off or for referencing past queries.

Current Build & References

Github Repo

References

We would like to sincerely thank our Industry Mentor, Dhaha Nur, as well as our Faculty Mentor, Rajesh Subramanyan for their guidance during this project. Also, our thanks to ACEP and the University of Washington for the opportunity to work on this project. Last but not least, to our TA Mentor, Rose Johnson for keeping us on track.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: DHAHA NUR, RAJESH SUBRAMANYAN, ROSE JOHNSON

SPONSOR: ALASKA CENTER FOR ENERGY AND POWER